UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI TECNICA E GESTIONE DEI SISTEMI INDUSTRIALI

Corso di Laurea in Ingegneria Meccatronica

Tesi di Laurea

Interplay between Communication and Synchronization in Networks of Neural Oscillators

Relatore
Ch.mo Prof. Giacomo Baggio

Candidato:
Andrea Pavan
Matricola: 1241762

Anno accademico 2021-2022

Abstract

In the present dissertation, we propose a simplified mathematical model of neural dynamics in order to investigate the interplay between synchronization and communication in biological neural networks. Specifically, starting from the physiological aspects of single neurons, we first examine two of the most renowned dynamical models used to analytically describe and characterize the functioning of neurons and neuronal populations. Then, we carefully analyze a two-dimensional oscillatory model of neural dynamics describing the firing rate activity of excitatory and inhibitory populations of neurons. Finally, the latter model is exploited to investigate the interplay between synchronization and communication in networks of neural oscillators, ranging from simple interconnection configurations to more general topologies.

Acknowledgements

First and foremost, I would like to express my deepest and sincere gratitude to my thesis supervisor, professor Baggio, for his invaluable patience and feedback, for his guidance and meticulous editing. Without his assistance and dedicated involvement in every step throughout the process, this work would have never been accomplished.

I would also like to thank my family, especially my mum, for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

Last but not least, thank you Steven, for always being there and supporting me. This thesis would have never been possible without you, your thoughtful presence and your caring affection.

Contents

Introduction

Consisting of billions of intricately connected neurons, the brain is arguably one of the most complex dynamical systems in nature, and understanding the dynamical laws that regulate brain behavior might reveal itself one of the most challenging and important problems of modern science. The first models of brain dynamics date back to the beginning of the 1900s, when Lapique [8] proposed a simplified electrical model for the generation of spikes in neurons (known as Leaky Integrate-and-Fire, or LIF, model). In 1952 Hodgkin and Huxley [6] proposed a more sophisticated conductance-based model of single neuron dynamics by relying on experimental data recorded from a squid giant axon. Building on these seminal works, computational neuroscientists have either focused on more realistic (yet less tractable) conductance-based models of single neuron dynamics or on analyzing the large-scale brain behavior arising from many interconnected simple (LIF-like) single-neuron models [2, 4]. In this thesis, we follow the second route and consider a simplified model of brain dynamics describing the activity of populations of excitatory and inhibitory neurons. Precisely, our model is a linear two-dimensional system where the states describe the (mean) firing rate activity of the excitatory and inhibitory populations. We then model large-scale brain activity as the interconnection of many of such excitatory-inhibitory (E-I, for short) systems.

An important feature of our model is that it supports oscillatory behaviors. Oscillations are ubiquitous features of brain dynamics and many works have investigated the role of oscillations in the brain, both from an experimental and theoretical viewpoint. In particular, some works have argued that oscillations are crucial for the efficient and flexible propagation of information between different brain areas, see, e.g., [1, 10], or for reliably processing and storing sensory information, see, e.g., [3]. Other works have linked the level of synchronization of oscillations between different brain regions to the

correct or pathological functioning of the brain, see e.g., [5, 9].

Building on our simplified networked model of brain activity, here we show that the level of synchronization between different E-I systems, as captured by the level of similarity of their natural frequencies, plays a key role in the information transmission performance of the brain, as quantified by the Signal-to-Noise ratio (SNR) at the receiver location.

Specifically, we provide both theoretical and numerical evidence that, if

1. the information content of the to-be-transmitted signal is concentrated around a single, sufficiently high frequency, and

2. the physical coupling between different E-I systems is dominated by excitatory connections,

then the best SNR performance is achieved when all E-I systems are characterized by the same natural frequency, i.e., they are completely synchronized. Although preliminary and based on simplifying assumptions, our analysis suggest that, to operate efficiently, the brain may require the activity of different brain regions to be synchronized.

The remainder of the thesis is organized as follows. The first part is mainly focused on delineating the simplified neural model that will be exploited to investigate the relationship between communication and synchronization in networks of neural oscillators: in the fist chapter we start by introducing the main physiological aspects correlated to the functioning of the neurons and the nervous system, describing how electrical signals are encoded in *action potentials* in order to be effectively carried through networks of neurons, even over large distances. Chains of actions potentials are usually referred to as *spike trains*, which usually convey information through the timing of the action potentials. Although spike trains are typically stimulus-dependent as well as trial-dependent and thus they must be treated probabilistically, they still can be characterized by the so-called *firing rate*. In particular, the definitions of the firing rate will be vastly outlined and explained along the chapter, where we will also provide some estimation methods of the firing rate.

Later on, the second chapter will present two of the most famous neural models, namely the *LIF model* and the *Firing Rate model*. The former is a renowned single-compartment model that describes the evolution of the membrane potential of the single neuron through a circuit-like representation, whereas the latter refers to populations of neurons and focuses on describing

the evolution of their mean firing rates. We will then attempt to establish a "link" between the two apparently distant models showing how they can be tightly correlated to each other by means of the definition of firing rate. We will finally derive our simplified *Excitatory-Inhibitory model* of neural dynamics by linearizing the equation of the Firing Rate model around an equilibrium point and assuming that the neuron populations are composed by either excitatory and inhibitory neurons in accordance with Dale's law. In particular, we will assume that the activity of each neural population can be described by two variables that encapsulate the averaged dynamics of the excitatory and inhibitory sub-populations. The end of the chapter will be largely dedicated to analyzing the Excitatory-Inhibitory model starting from computing the eigenvalues of its state matrix and ending up into the careful analysis of its frequency response function: particular emphasis will be put on the role of the *natural frequency* of the E-I oscillator and on highlighting the conditions that allow us to exploit the resonating behaviour of the model. Despite its simplicity, the Excitatory-Inhibitory (E-I) model will be able to reveal its effectiveness in the second part of the thesis.

The spotlight of the second part of thesis will be mainly centered on investigating the relationship between communication and synchronization in networks of neural oscillators as defined with our simplified E-I model. We will start by outlining our *communication framework*, in which we assume that the propagation of the signal is carried out through a *noisy channel*, which returns a corrupted output signal. With this in mind, we will then introduce the main *metric* that will be used to quantify the level of quality of the signal propagation throughout the network, namely the *Signal-to-Noise ratio*, shortly SNR. As we will show, higher values of SNR will result into a better communication performance of the channel: as a result, we will make use of this relationship in order to find proof that the frequency synchronization of the nodes actually results in higher values of the SNR. The third chapter concludes with the formulation of the *optimization problem* that will be thoroughly tackled in the following chapters: we will be interested in finding which values of the natural frequencies actually correspond to the best propagation of the input signal possible, that is the maximum value of the SNR, aiming to understand whether the frequency synchronization of the E-I oscillators implies a better propagation of the input signal (when concentrated on a single high frequency) or not.

The rest of the second part is dedicated to solving the optimization problem for several configurations and topologies of the E-I networks: starting from

the single oscillator dynamics and the simplest interconnections of two E-I oscillator, i.e., the series and the positive feedback configurations, we will finally analyze more complex configurations with multiple oscillators. More specifically, the last part will be focused on more general interconnection patterns based on *directed acyclic graphs*, shortly DAG. We will prove that the synchronization of the nodes actually benefits the propagation of the signal, under accurate circumstances, giving insights on why communication is actually favored by having synchronized E-I systems in the presence of a majority of excitatory connections, rather than inhibitory ones.

Lastly, we will discuss the intrinsic limits of the thesis and some future research directions aiming at better understanding the functioning and the underlying complexity of neural clusters, as well as the role of synchronization in communication-oriented networks.

Part I

Neurons and Mathematical Models

Chapter 1

The Neuron

In this chapter we analyze the importance of the physiology of the neuron and the topology of neuronal networks in order to propagate signals rapidly over large distances. We start by describing the physiological structure of a neuron and the main features of a neuronal network. Then we focus on the mechanism that underlies the generation of the action potentials, i.e., the spikes that can travel down nerve fibers carrying the information. We then introduce the concept of "firing rate", which is the central focus of the chapter and thus it will be thoroughly investigated. Finally, we present a way to experimentally estimate the "firing rate" through different trials.

1.1 Physiological aspects

Neurons (also called nerve cells) are the fundamental units of the brain and nervous system, the cells responsible for sending motor commands to our muscles for receiving sensory inputs from the external world and for processing the electrical signals in our brain.

Neurons are embedded in a network of billions of other neurons and glial cell (i.e., non-neuronal cells that provide physical and metabolic support to neurons) that form the brain tissue. The brain is organized in different regions and areas. The cortex can be thought of as a thin but extended sheet of neurons, folded over other brain structures. Some cortical areas are mainly involved in processing sensory inputs, other areas are involved in working memory or motor control.

Neurons are typically classified into three types based on their functions: *sensory neurons*, which respond to stimuli of sensory organs and send signals to the spinal cord or the brain; *motor neurons*, which control different tasks, from muscle contractions to glandular output; *interneurons*, which connect neurons to other neurons within the same region of the brain.

As shown in Fig. 1.1, a typical neuron can be divided into three functionally distinct parts, called *soma*, *dendrites* and *axon*: the soma is the cell body, the dendrites are the extensions of the nerve cell that allow the propagation of the electrical signal, and the axon is a cable-like projection that carries nerve signals away from the soma and back to it. The typical branching structure of the dendritic tree allows a neuron to receive inputs from many other neurons through the synaptic connections located at the end of the dendrites.

Roughly speaking, the neuron can be thought of as a simple processing device. The dendrites play the role of the 'input device', which collects signals from other neurons and transmits them to the soma. The latter is the 'central processing unit' that performs an important non-linear processing step: if the sum of the inputs arriving at the soma exceeds a certain threshold, then an output signal is generated. The output signal is taken over by the 'output device', the axon, which delivers the signal to other neurons.

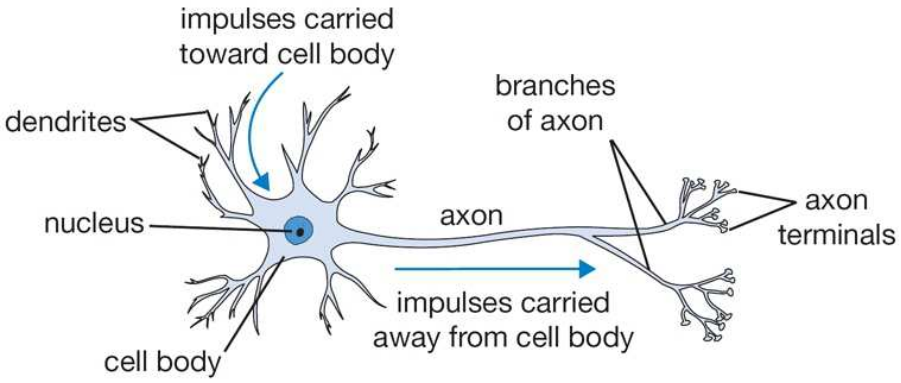


Figure 1.1: Schematic structure of a neuron cell.

A group of connected neurons is also called a *neural circuit*, i.e., a population of nerve cells interconnected by synapses to carry out a specific function when activated. Neural circuits interconnect to one another to form large scale brain networks.

Neurons communicate with each other via synapses which can be *excitatory*

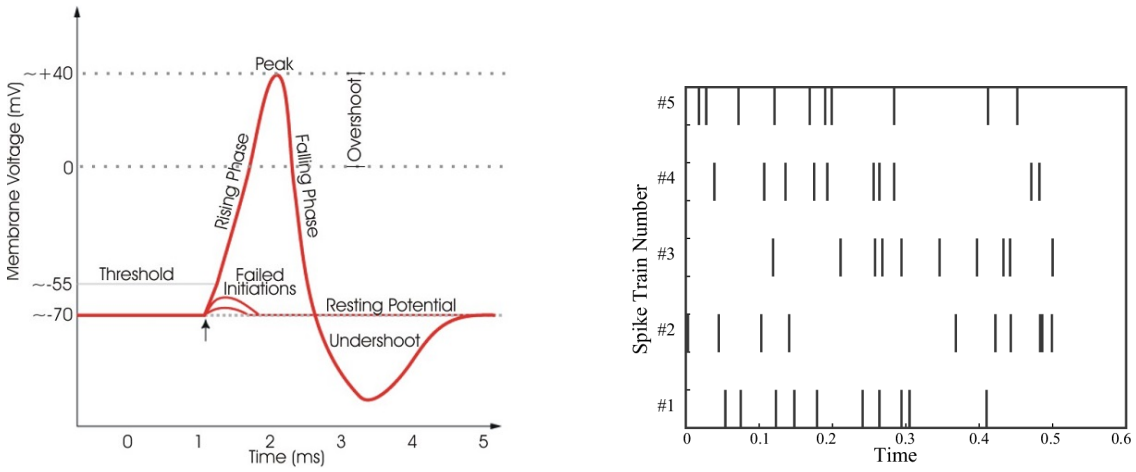
or *inhibitory*, either increasing or decreasing the activity in the target neuron, respectively. In particular, from a physiological point of view, the cell membrane of the axon and soma contain *voltage-gated ion channels*, that allow ions, predominantly sodium (Na^+), potassium (K^+), calcium (Ca^{2+}) and chloride (Cl^-), to move in and out of the cell. This particular flow of ions across the membrane is controlled by the gates in response to voltage changes and to both internal and external signals.

1.2 Action Potential

The electrical signal of relevance to the nervous system is the difference in electrical potential between the interior of the neuron and the surrounding extracellular medium. Under resting condition the potential inside the cell membrane is kept near to -70 mV (relative to the potential of the surrounding bath, conventionally defined to be 0 mV). In this scenario, the cell is said to be *polarized*. Ion pumps located in the cell membrane maintain the concentration gradient that supports this potential difference.

Current in the form of positively charged ions flowing out of the cell through open channels makes the membrane potential more negative, i.e. process of *hyperpolarization*. On the contrary, current flowing into the cell reduces the membrane potential even to positive values, i.e. process of *depolarization*. If a neuron is depolarized sufficiently to raise the membrane potential above a specific threshold level, the neuron generates an *action potential*, which is a fluctuation of the membrane potential that has an amplitude of about 100 mV and typically a duration of about 1-2 ms, as shown in Figure 1.2a. Thus, the action potential corresponds to a voltage pulse or spike.

Action potentials are of great importance because they are the only form of membrane potential fluctuation that can propagate over large distances, while sub-threshold potential fluctuations are severely attenuated over distances. In fact, the shape of electrical pulse, as shown in Figure 1.2a, does not change as the action potential propagates along the axon. A chain of action potentials emitted by a single neuron is generally called a spike train, namely, a sequence of stereotyped events which occur at regular or irregular intervals. An example of a set of different spike trains is shown in Figure 1.2b where the vertical axis shows the number of trials and the horizontal axis represents the time. Each bar represents a single action potential occurring at a precise time. Since isolated spikes of a given neuron look alike,



(a) Action potential (b) Spike trains

Figure 1.2: In Figure 1.2a the typical shape of an action potential is shown: starting from a resting potential, when the potential crosses a particular value, called threshold potential, the neuron spikes and the potential reaches its peak value. After the spike, the potential falls below the resting potential with an undershooting trajectory and it relaxes again to the resting potential during its refractory period. In Figure 1.2b a different set of spike trains during multiple trials is shown. In particular, each bar represents a single action potential.

the form of the action potential does not carry any information. Rather, it is the number and the timing of spikes which matter. It is clear that the action potential is basically the elementary unit of signal transmission. Action potentials in a spike train are usually well separated. Even with very strong inputs, it is impossible to excite a second spike during or immediately after a first one. The minimal distance between two spikes defines the absolute refractory period of the neuron. The absolute refractory period is followed by a phase of relative refractoriness where it is difficult, but not impossible to excite an action potential.

1.3 Firing rate

1.3.1 Spike-count rate

Action potentials usually convey information not through their shape, but through their timing and are typically treated as identical stereotyped events.

Thus, an action potential sequence can be characterized as a sum of infinitesimally narrow, idealized spikes in form of the Dirac function, namely the δ function,

$$\rho(t) = \sum_{i=1}^{n} \delta(t - t_i). \tag{1.1}$$

In Equation (1.1) $\rho(t)$ represents the *neural response function*, i.e., the neural activity, which is used to express sums over spikes. Note that counting the number of spikes corresponds to compute integrals over time of the neural response function, since the δ function has the fundamental property that

$$\int_{-\infty}^{\infty} f(x)\delta(x - x_0)dx = f(x_0)$$

and so, assuming that $f(x) = 1$, we can write

$$\int_{-\infty}^{\infty} 1 \cdot \delta(x - x_0)dx = \int_{-\infty}^{\infty} \delta(x - x_0)dx = 1.$$

Neural responses can typically vary from trial to trial even when the same repeated stimulus is presented, due to the multitude of sources of variability that can affect neural firing, such as randomness associated with biophysical processes and the effects of other cognitive processes happening during the trial. The overall complexity of characterizing the relationship between stimulus and neural response due to the trial-to-trial variability of action potential sequences make it impossible to be able to describe and predict the timing of each spike deterministically.

Therefore, neuronal responses are typically treated probabilistically, meaning that we need to introduce a quantity that can take into account and eventually describe the probabilities that different spikes are evoked by specific stimulus. Thus, the sequences of action potentials might be characterized by a specific *firing rate*. This term is generically applied to different quantities, but the simplest one might be the *spike-count rate*, namely the time average of the neural response function over the duration of the trial,

$$r^{sp} = \frac{n}{T} = \frac{1}{T}\int_0^T \rho(\tau)d\tau. \tag{1.2}$$

where $n = \int_0^T \rho(\tau)d\tau$ is the number of spikes over the interval $[0, T]$, which is the duration of the trial.

1.3.2 Time-dependent firing rate

The spike-count rate can be determined in a single trial, but this would mean losing temporal resolution about the variations of the neural response. The solution is to average over multiple trials, defining a *time-dependent firing rate* as the trial-averaged number of spikes appeared during a short interval between t and $t + \Delta t$, divided by the duration of the interval.

The time-dependent firing rate for K different trials is given by

$$r(t) = \frac{1}{\Delta t} \int_{t}^{t+\Delta t} \langle \rho(\tau) \rangle d\tau \tag{1.3}$$

where $\langle \rho(\tau) \rangle = \frac{1}{K} \sum_{i=1}^{K} \rho_i(t)$ is the trial-averaged neural response function, namely the average over all the trials of the neural responses $\rho_i(t)$ obtained exploiting the same stimulus. Note that the value of Δt must be large enough so there are sufficient spikes within the interval to obtain a reliable estimate of the average.

The definition of the time-dependent firing rate shown in Equation (1.3) is particularly relevant because $r(t)\Delta t$ can also represent the probability of a spike occurring during a short interval of duration Δt around the time t.

As an experimental procedure, the time-dependent firing rate measure is a useful method to evaluate the neuronal activity, in particular in the case of time-dependent stimuli. The obvious problem with this approach is that it cannot be the encoding scheme used by neurons in the brain: neurons cannot wait for the stimuli to repeatedly occur in the exactly same manner before generating a response. Moreover, the dynamics of many environmental signals are measured in milliseconds and, during these milliseconds, neurons can only fire once or twice, due to their intrinsic refractoriness. With such a number of spikes, it seems impossible to encode the signal by their average rate. Hence, the central problems with the firing rate approach are that not all interesting experiments can be forced into a repeated-trial design and that averaging across trials requires stationarity across trials, which cannot be always guaranteed.

These particular issues lead to several questions: can the firing rate be seen as a sufficiently good description of the neural activity? Is it then possible to reduce the spiking interactions of neurons to the interaction of firing rates?

1.3.3 Neural Coding

Neural coding describes the study of information processing by neurons and particularly the transduction of environmental signals and internal signals of the body into neural activity patterns. As we have discussed, much is known about the biophysics aspects of neural responses, i.e. how a spike is generated and how the signal transmission takes place at the neurons level. However, the impact of series of spikes is not well known, and it is even less clear how the information is encoded and how that information is utilized in subsequent processing stages.

Moreover, the nature of the neural code is a topic of intense debate within the neuroscience community. Indeed it is still not clear whether neurons use rate coding or temporal coding to encode information.

Usually information is carried by single neurons, but is typically encoded by neuronal populations. When studying population coding, it is necessary to consider whether the neurons are uncorrelated with each other, meaning that they act independently, or whether the correlations between the neurons can actually carry additional information. When analyzing the population code it is common to consider the response of each neuron statistically independent, meaning that the spike trains can be combined without taking into account the correlations between the single spikes.

1.3.4 Firing Rate Estimation

Trying to exactly estimate the time-dependent firing rate is not an easy task and this difficulty is especially due to the lack of data available from a finite number of trials. Moreover there is not a unique expression that approximates the firing rate $r(t)$, thus many methods can be presented to give an estimate of $r(t)$.

Here we assume that the data from n spikes trains has been recorded independent and statistically identical trials. Let $(0, T]$ denote the duration of the trials and let $0 \leq t_1 < t_2 < \cdots < t_n \leq T$ be the spike times for a particular train of spikes. We will denote the estimated firing rate with $\hat{r}$.

One of the oldest methods of dealing with time-dependent firing rate exploits the concept of "instantaneous firing rate" where the reciprocal values of the interspike intervals (also called ISI) are used to determine the firing rate. The instantaneous firing rate for a given spike train $t_1, t_2, \ldots, t_n$, at a time t

is estimated as:

$$\hat{r}(t) = \frac{1}{t_i - t_{i-1}}, \qquad t_{i-1} \leq t < t_i. \tag{1.4}$$

In order to obtain an approximate estimate of the firing rate from a few spike trains, time averaging over short time intervals is performed. This process is called *binning* and it is used in various firing rate estimation methods. For this particular method, n spike trains under independent and identical statistical trials are taken and superimposed. The observation time T is divided into N_b discrete time bins of width Δt and the number of spikes in the ith bin, across all trials, is denoted by Σ_i. Thus, the firing rate $\hat{r}(t)$ for the ith bin is estimated as

$$\hat{r}(t) = \frac{\Sigma_i}{n\Delta t}, \quad \text{for } t \in [(i-1)\Delta t, \Delta t]. \tag{1.5}$$

The same process is repeated for each one of the N_b bins to compute a piece-wise constant function (similar to a time histogram) representing the estimated firing rate. As a consequence, the estimated firing rate $\hat{r}(t)$ will be quantized and although this method is very easy to implement, it might not portray the firing rate fluctuations accurately among all the trials. In particular, the width Δ of each bin is a critical parameter: decreasing Δt would mean increasing the temporal resolution of the resulting firing rate, but this would also mean having a loss of resolution for distinguishing different rates.

In order to avoid a quantized firing rate, we could instead take a moving average obtained through a sliding window of duration Δt which slides along the spiking train counting at each step the number of spikes that are within the window. Therefore, for spike time data $t_1, t_2, \ldots, t_N$ the firing rate $r(t)$ will be approximated as the sum of a window function over the times t_i,

$$\hat{r}(t) = \sum_{i=1}^{N} \kappa_\Delta(t - t_i) \tag{1.6}$$

where the window function $\kappa_\Delta(t)$ with bandwidth Δ is

$$\kappa_\Delta(t) := \begin{cases} 1/\Delta & \text{if } -\Delta/2 \leq t \leq \Delta/2 \\ 0 & \text{otherwise.} \end{cases} \tag{1.7}$$

Mathematically, this is equivalent to the convolution of the data with the

window function, so Equation (1.6) can be rewritten for $\Delta \to 0$ as the following convolution integral

$$\hat{r}(t) = \int_{-\infty}^{\infty} \kappa_\Delta(\tau)\rho(t-\tau)d\tau, \tag{1.8}$$

where $\rho(t)$ is the neural response and $\kappa_\Delta(t)$ is the window function, where Δ is the bandwidth or the so-called *smoothing parameter*. In particular, the convolution integral in Equation (1.8) can be seen as the output of a linear system with input $\rho(t)$ and impulse response $\kappa_\Delta(t)$, where the latter is typically referred to as *filter kernel* or *kernel function*. The kernel function must satisfy these condition:

$$\kappa_\Delta(t) \geq 0, \tag{1.9a}$$

$$\int_{-\infty}^{\infty} \kappa_\Delta(\tau)d\tau = 1, \tag{1.9b}$$

$$\int_{-\infty}^{\infty} \tau\kappa_\Delta(\tau)d\tau = 0. \tag{1.9c}$$

Instead of a rectangular-shaped sliding window, the window function could be a Gaussian function, whose plot is presented in Figure 1.3a:

$$\kappa_\sigma^\gamma(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{\left(-\frac{t^2}{2\sigma^2}\right)\right\} \tag{1.10}$$

where the parameter σ rules the temporal resolution of the approximated firing rate $\hat{r}(t)$. Using this particular filter kernel, which is a continuous window function, allows us to obtain a smoother firing rate.

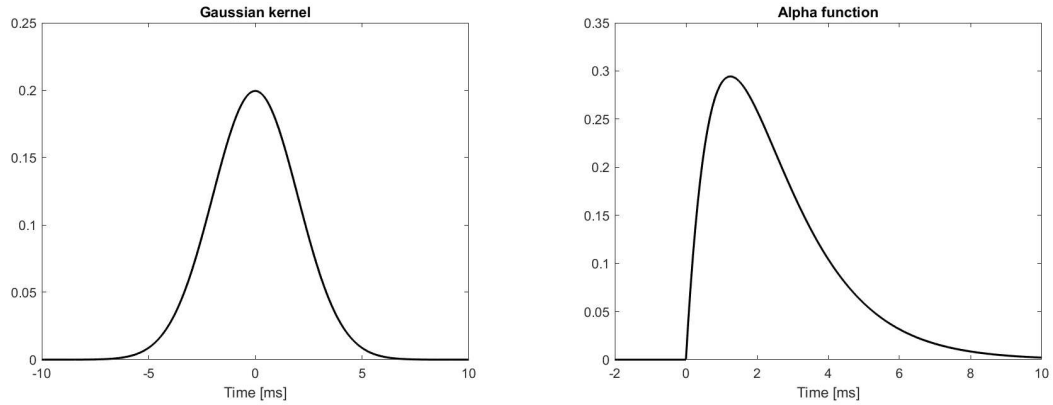
Eventually, in order to take into account that each postsynaptic neuron has access at time t only to the spikes that occurred before t, we could introduce a *causal* window function, like the α-function, as exemplified in Figure 1.3b:

$$\kappa_\alpha^{exp}(t) = [\alpha^2 t \exp\{(-\alpha t)\}]_+ \tag{1.11}$$

where by $[x]_+$ we denoted the function

$$[x]_+ = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

For the alpha function the temporal resolution of the estimated firing rate $\hat{r}(t)$ is given by the parameter $1/\alpha$.



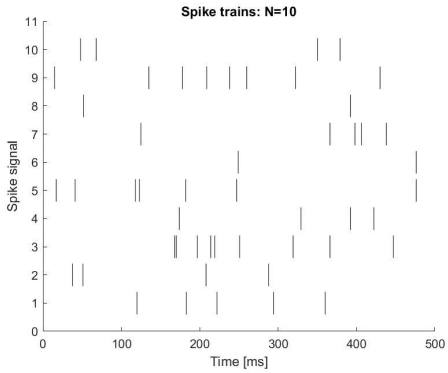
(a) Gaussian Kernel (b) Alpha function

Figure 1.3: Figure 1.3a shows the plot of the gaussian kernel, where we set $\sigma = 2$. Figure 1.3b shows the plot of the alpha function. where we set $\alpha = 0.8$. In this last case, it is possible to notice that the plot is asymmetrical, since the function is causal.

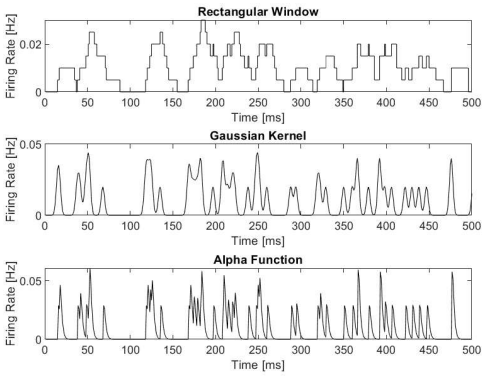
The choice of the free parameter Δ is a crucial problem with the kernel smoothing method since it affects the smoothness of the firing rate estimate and highlights the temporal structure of the underlying spiking activity. There are many ways to compute an optimized kernel bandwidth in order to minimize the error between the estimate $\hat{r}(t)$ and the firing rate $r(t)$. Further details can be found for example in [11].

Example of estimation of the Firing rate

In this section we show the different estimated firing rate plots obtained using the different kernel functions described and shown above. In particular, we start from a set of N trials, where each trial is characterized by a different spike train (plotted with a set of bars representing the sequence of spikes). Each spike train is generated by means of an apposite Matlab function: we suppose that the generation of the spikes can be modelled as a Poisson process with rate (or intensity) λ, and thus we obtain the set of N spike trains and duration T, as shown in Figure 1.4a. We then filter the data with the different window functions (respectively, the rectangular window function, the Gaussian kernel and the alpha function) to estimate the firing rate of the N different trials and obtain an estimation of the firing rate through the trials. The results are shown in Figure 1.4b.



(a) Set of spike trains during N trials.



(b) Firing rate estimation with different filter functions

Figure 1.4: In Figure 1.4a our set of $N = 10$ spike trains is shown: each bar represents a spike occurring at a precise time. The spiking events are modelled as a Poisson process with rate $\lambda = 10$. Figure 1.4b shows the results of the estimation of the firing rate.

The code used to generate the spike trains and the plots can be found in Appendix C.

Chapter 2

Neural Models

When modeling neurons, there are two types of complexity that must be taken into account: the intricate and rich dynamics of the neurons and the elaborate morphology of the neural networks, which allows neurons to receive and integrate inputs from so many other neurons. Neural models range from oversimplified models to highly detailed descriptions, involving lots of differential equations. It is obvious that oversimplified models can give misleading results, but they generally help to understand the underlying structure and behaviour when handling intricate networks of neurons. There should be a clear trade-off between the amount of detail that can be devoted to modeling each neuron (or even each synapse) and the size of complexity of the network that can be reached.

The aim of this text is to present two of the simplest, yet effective, models that can describe the neuron *per se* and consequently the network in which neurons are interconnected: the *integrate-and-fire model*, also known as *spike-based model*, and the *firing-rate model*. Integrate-and-fire models are generally used to describe the dynamics of single neurons. This particular model allows simulations of networks with even thousands of neurons, thanks to its simplicity, and it attempts to describe the single neuron starting from its biophysical behaviour.

On the other hand, firing-rate models avoid the time scale dynamics required to simulate action potentials and are much easier to simulate and run on computers. They also allow to treat some aspects of network dynamics that could not be treated in the case of spike-based models and can actually be used to build simplified versions of the network by means of "averaging units",

so that the output of the model unit is simply the average of the firing rates of the neurons that the model collectively represents. However, one of the firing-rate models limitations is that they are usually restricted to cases where the firing of neurons in a network is uncorrelated and with little synchronous firing.

2.1 Integrate-and-Fire Models

To a first approximation, the neuronal dynamics can be conceived as a summation (or "integration") process combined with a mechanism that triggers action potentials above some threshold voltage. In order to build a phenomenological model of neuronal dynamics, we describe the critical voltage for spike initiation by a formal threshold V_{th}. If the voltage $V(t)$ (that contains the summed effect of all inputs reaching the neuron) reaches V_{th} from below, we say that the neuron spikes. The moment of threshold crossing defines the firing time t_f.

This model exploits the fact that neuronal action potentials of a given neuron usually have the same shape. Under this basic assumption, it is clear that the shape cannot be used to transmit information: rather information is contained in the presence or absence of a spike. Thus, action potentials are reduced to simple 'events' that happen at a precise moment in time.

Neuron models where action potentials are described as events are called 'Integrate-and-Fire' models. Generally speaking, integrate-and-fire models usually comprise two separate components which are apt to describe the dynamics of the spiking neuron: first, a differential equation that describes the evolution of the membrane potential $V(t)$; and second, an intrinsic mechanism through which the neuron generates spikes when crossing a threshold value. In the next section we will derive them considering the biophysical behaviour of a single neuron.

2.1.1 'Leaky Integrate-and-Fire' Model

The term *integrate-and-fire model* usually refers to many different models, but here we will introduce and describe the simplest one in the class of spike-based models, the so-called 'Leaky Integrate-and-Fire' model, also called "LIF model".

As mentioned in the previous chapter, a neuron typically fires an action

potential when its membrane potential $V(t)$ reaches a threshold value V_{th} which usually is in the range between -55 and -50 mV. During the action potential, the membrane potential $V(t)$ follows a stereotyped trajectory and the returns to its hyperpolarized value, relative to the threshold potential V_{th}.

Therefore, the integrate-and-fire models exploits this simple mechanism: an action potential, which basically can be perceived as a spike, occurs whenever the membrane potential of the neuron reaches a threshold value V_{th}. After the action potential, the potential is instantaneously set to a precise reset value V_{reset}. Despite its simplicity, due to the absence of further biophysical descriptions, the integrate-and-fire is still an extremely powerful description of the neuron activity.

In particular, the LIF model is a so-called *single-compartment model*, meaning that the membrane potential of a single neuron can be described by a single variable $V(t)$, in contrast with multi-compartment models which can also describe the spatial variations of the membrane potential. In general, the cell membrane creates a capacitance C_m (which typically varies from 0.1 to 1 nF) and can be characterized by a membrane resistance R_m (which might vary considerably among cells, but it is typically in the range from 10 to 100 MΩ). The product of the membrane capacitance C_m and the membrane resistance R_m is a quantity called membrane time constant $\tau_m = R_m C_m$ which typically is in the range 10 to 100 ms.

Equation (2.1) shows the dynamics of the membrane potential $V(t)$ for a single-compartment model, stating that the rate of change of the membrane potential $V(t)$ is proportional to the total amount of current entering the neuron:

$$c_m \frac{dV(t)}{dt} = -i_m + \frac{I_e}{A}. \tag{2.1}$$

The membrane current i_m[1] is usually characterized as a current per unit area, hence in Equation (2.1) A denotes the surface area of the neuron, $c_m = C_m/A$ is the specific membrane capacitance, while I_e denotes an eventual external current injected into the cell, for example through an electrode.

[1] Each ion channel i of the membrane can be characterized by a specific conductance g_i and with a reversal potential, also called Nernst potential, E_i, such that when $V(t) = E_i$ there is no flow of current through the channel i. The total membrane current i_m per unit area can be obtained by summing the currents over the different channels: $i_m = \sum_i g_i(V(t) - E_i)$.

The structure of such Equation recalls the model of an electrical circuit, the so-called *equivalent circuit*, consisting of a single capacitor and a set of conductances respectively representing the cell membrane capacitance and the ion channels in the cell membrane.

In the simplest version of the LIF model, all the active conductances of the ion channels are ignored and the entire membrane conductance $g_m = 1/r_m$ is modeled as a passive leakage term, $i_{m,lk} = g_{lk}(V(t) - V_{rest})$, where g_{lk} is the so-called *leakage conductance*.

This leads to the following dynamics:

$$\begin{cases} c_m \dfrac{dV(t)}{dt} = -g_{lk}(V(t) - V_{rest}) + \dfrac{I_e}{A}, & V(t) < V_{th}, \\ V(t) = V_{reset}, & V(t) \geq V_{th}, \end{cases} \tag{2.2}$$

where $V(t)$ is the momentary value of the membrane potential, namely the momentary value of the action potential of the neuron, $I_e(t)$ the input current from a possible external source, and V_{rest}, V_{reset}, V_{th} the resting, reset, threshold potential, respectively.

By multiplying Equation (2.2) by the specific membrane resistance r_m, we can rewrite the basic Equation (2.3) of the leaky integrate-and-fire model as follows:

$$\begin{cases} \tau_m \dfrac{dV(t)}{dt} = -(V(t) - V_{rest}) + R_m I_e(t), & V(t) < V_{th}, \\ V(t) = V_{reset}, & V(t) \geq V_{th}, \end{cases} \tag{2.3}$$

where $R_m = r_m/A$ is the membrane resistance, $\tau_m = r_m c_m = R_m C_m$ is the membrane time constant.

According to this model, whenever $V(t)$ reaches the threshold value V_{th} an action potential, i.e., a spike, is fired and the potential $V(t)$ is reset automatically to V_{reset}. The equivalent electric circuit is depicted in Figure 2.1.

According to this model, whenever $V(t)$ reaches the threshold value V_{th} an action potential, i.e. a spike, is fired and the potential $V(t)$ is reset automatically to V_{reset}.

Equation (2.3) also indicates that when $I_e = 0$ the membrane potential $V(t)$ relaxes exponentially with time constant τ_m to its rest value V_{rest}. As we mentioned before, for a typical neuron the membrane time constant τ_m is

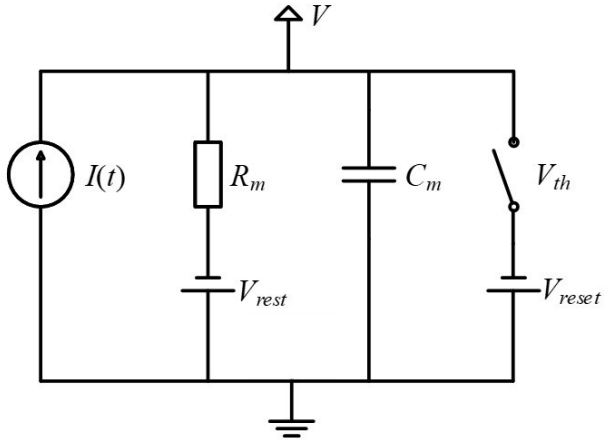


Figure 2.1: Equivalent model of a LIF model.

approximately around 10 ms, and hence rather relevant compared to the duration of a spike, which is approximately on the order of 1 ms.

In addition, the firing rate of a LIF model can be computed analytically with constant input current I_e: in fact, the subthreshold potential $V(t)$, meaning the value of the potential $V(t)$ under the threshold value V_{th}, can be determined by solving Equation (2.4), which is the solution of the differential Equation (2.3) for $I_e(t) = I_e$:

$$\begin{aligned} V(t) &= V_{rest} + R_m I_e + (V_0 - V_{rest} - R_m I_e)\, e^{-\frac{t}{\tau_m}} \\ &= V_\infty + (V_0 - V_\infty)\, e^{-\frac{t}{\tau_m}}, \end{aligned} \tag{2.4}$$

where we denoted with V_0 the initial value of the membrane potential $V_0 = V(t = 0)$ and with $V_\infty = V_{rest} + R_m I_e$ the equilibrium potential.

The subthreshold potential always follows the same trajectory, therefore we can assume that at $t = 0$ the neuron is at the reset potential, so that $V_0 = V_{reset}$ and that the next action potential will occur when the membrane potential reaches the threshold value V_{th} at time $t = t_{isi}$.

In case of a constant input current I_e, the interspike interval t_{isi}, which is defined as the time it takes to a membrane potential to reach the fixed threshold value V_{th} can be computed from Equation (2.4) as a function of I_e

$$V_{th} = V_\infty + (V_{reset} - V_\infty)\, e^{-\frac{t_{isi}}{\tau_m}}. \tag{2.5}$$

We can then solve the Equation for t_{isi}:

$$t_{isi} = \tau_m \ln\left(\frac{V_\infty - V_{reset}}{V_\infty - V_{th}}\right) = \tau_m \ln\left(\frac{R_m I_e + V_{rest} - V_{reset}}{R_m I_e + V_{rest} - V_{th}}\right). \tag{2.6}$$

A finite value of t_{isi} is obtained only if $R_m I_e + V_{rest} - V_{th} \geq 0$, meaning that only currents $I_e \geq (V_{th} - V_{reset})/R_m$ produce spikes. This can be easily seen in Figure 2.2 where the evolution of the potential $V(t)$ for different values of input constant current I_e is shown. The plots were generated with an apposite Matlab code, which can be found in Appendix C.

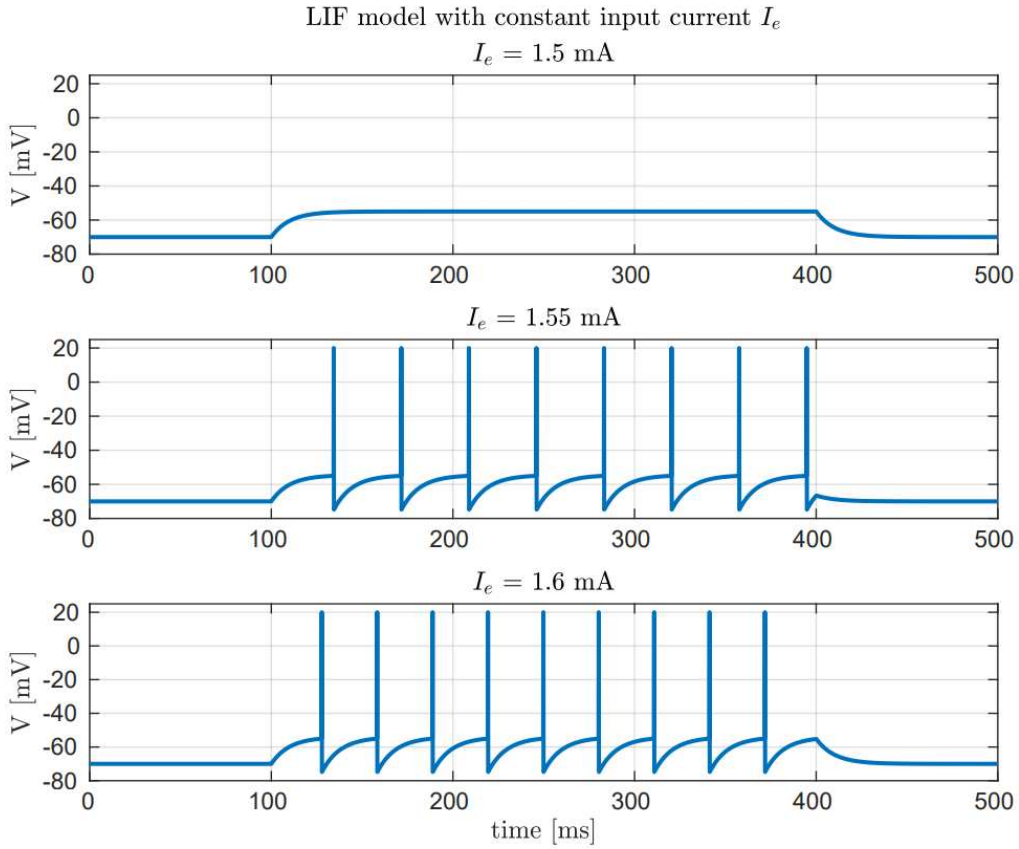


Figure 2.2: Plots of the potential $V(t)$ for different values of constant input current I_e. It is possible to see that, in order to spike, the neuron has to receive an input current $I_e > 1.5\,mA$.

By exploiting the approximation $r \approx \frac{1}{t_{isi}} = r_{isi}$, it is possible to determine

the firing rate r:

$$r \approx r_{isi} = \frac{1}{t_{isi}} = \left(\tau_m \ln\left(\frac{R_m I_e + V_{rest} - V_{reset}}{R_m I_e + V_{rest} - V_{th}}\right)\right)^{-1} =: f(I_e). \tag{2.7}$$

The last equation is valid if $R_m I_e > V_{th} - V_{reset}$, otherwise $r(t) = 0$. The function $f(\cdot)$ is called *activation function*, which basically is the function that relates the input current to the firing rate of the neuron in the steady-state regime. The plot of the activation function is shown in Figure 2.3.

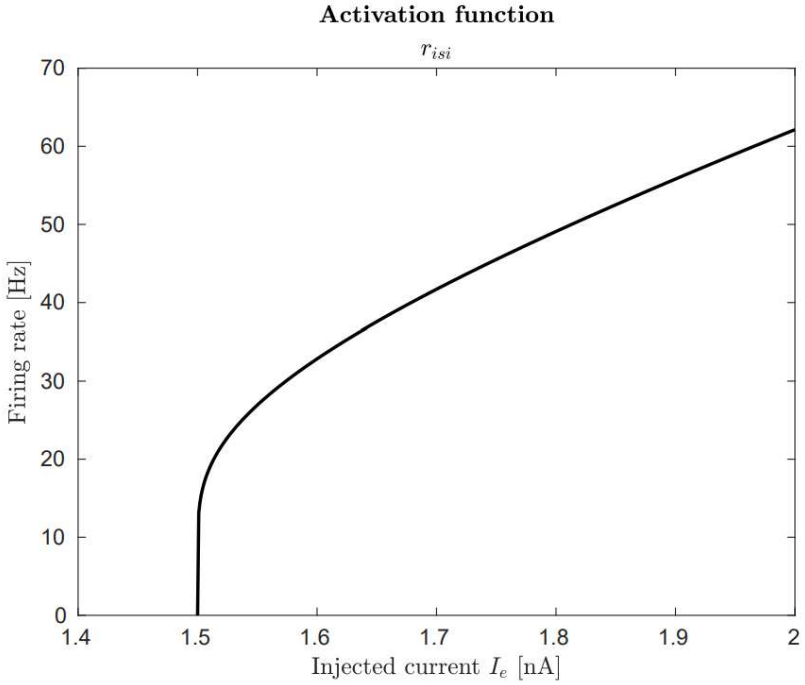


Figure 2.3: Plot of the activation function with the following parameters $R_m = 10\,M\Omega$, $\tau_m = 10\,ms$, $V_{reset} = -75\,mV$, $V_{rest} = -70\,mV$, $V_{th} = -55\,mV$, $V_{spike} = 20\,mV$.

Seeking an approximate formula for the interspiking interval t_{isi}, we can write

$$t_{isi} = \tau_m \ln\left(\frac{R_m I_e + V_{rest} - V_{th} + V_{th} - V_{reset}}{R_m I_e + V_{rest} - V_{th}}\right) = \tau_m \ln\left(1 + \frac{V_{th} - V_{reset}}{R_m I_e + V_{rest} - V_{th}}\right)$$

and exploiting the approximation $\ln(1 + x) \approx x$, for sufficiently small values of x, we can obtain, for sufficiently large values of I_e

$$t_{isi} \approx \frac{\tau_m (V_{th} - V_{reset})}{R_m I_e + V_{rest} - V_{th}},$$

and consequently for the firing rate

$$r \approx \frac{1}{t_{isi}} \approx -\frac{V_{th} - V_{rest}}{\tau_m(V_{th} - V_{reset})} + \frac{1}{C_m(V_{th} - V_{reset})} I_e = \alpha + \beta I_e, \tag{2.8}$$

which gives us a linear approximation of the function $f(I_e)$ displayed in Equation (2.7) and shows that r grows linearly with I_e, for sufficiently large values of I_e.

In case of a non-constant yet slowly time-varying input current $I_e(t)$, we can assume that the firing rate will eventually follow the fluctuations with an intrinsic delay, therefore it is legit to model the firing rate through the following low-pass filter equation

$$\tau_r \frac{dr(t)}{dt} = -r(t) + f(I_e(t)), \tag{2.9}$$

where τ_r is the firing rate constant, which describes how rapidly the firing rate r reaches its steady-state value. A similar result will appear in the next section, in which we will derive the basic equation of a firing rate model. For further details on the LIF model and its properties, see, e.g., [2].

2.1.2 Validity of the model

The leaky integrate-and-fire model is an extremely simplified neuron model and we cannot expect it to explain the complete biochemistry and biophysics of neurons. However, the integrate-and-fire model is surprisingly accurate when it comes to generating spikes, i.e., precisely timed events in time. Thus, it could potentially be a valid model of spike generation in neurons, or more precisely, in the soma.

As discussed above, neurons not only show refractoriness after each spike but also exhibit adaptation which builds up over hundreds of milliseconds. A simple leaky integrate-and-fire model does not perform well at predicting the spike times of a real neuron. However, if adaptation (and refractoriness) is added to the neuron model, the prediction seems to work quite well.

As we will see, it is possible to make the transition from single-neuron models to large and structured populations. This does not mean that we can understand the full structure of the brain, but understanding the principles of large populations of neurons from simplified neuron models is a first and important step in this direction.

2.2 Firing-Rate Models

Firing-rate models are often used when studying networks or populations of neurons since it does not consider the behaviour of the single neuron, in particular the action potential as an "event", but the overall firing rates of the network of N neurons.

As we discussed before, the sequence of spikes generated by a neuron is completely characterized by the neural response function $\rho(t)$, which consists of δ function spikes located at times when the neuron fired action potentials. In firing-rate models the description of a spike sequence provided by the neural response function $\rho(t)$ is replaced by the approximate description provided by the firing rate $r(t)$. This replacement is justified by the assumption that each neuron of the network has a large number of inputs. The replacement is also possible if we assume that the quantities of relevance for network dynamics are relatively insensitive to the trial-to-trial fluctuations in the spike sequences represented by $\rho(t)$.

Indeed, for any single synaptic input, the trial-to-trial variability is likely to be quite large. However, for uncorrelated presynaptic spike trains, using presynaptic firing rates in place of the actual presynaptic spike trains may not significantly modify the dynamics of the network. On the other hand, the firing rate model will fail to describe a network where the presynaptic spike trains are correlated: this could happen for example when the neurons spike synchronously.

2.2.1 Derivation of the model

In a firing-rate model, each neuron i can be described at time t by a firing rate $r_i(t)$, where $i = 1,2,...,N$ and N is the number of neurons in the network. As shown in Equation (2.7), each firing rate relaxes with a time-constant τ_r to a steady-state value given by the *activation function* f, which describes the relationship between firing rate and input current for the neuron.

The total input current $I(t)$ for a neuron i is the sum of the input $I_{e,i}(t)$ from sources outside the network, such as sensory input, and a term describing the total synaptic input $I_{s,i}(t)$ reaching the neuron i from other neurons of the same network. Building on the firing rate analysis of the LIF dynamics and, in particular, on Equation (2.7), the resulting dynamics for the firing

rate $r_i(t)$ of the i-th neuron is taken to be of the form:

$$\tau_r \frac{dr_i(t)}{dt} = -r_i(t) + f\left(I_{e,i}(t) + I_{s,i}(t)\right) \tag{2.10}$$

where τ_r is the firing rate time constant, which characterizes the time scale of variation of the firing rate, and the function f is the so-called activation function which is typically taken to be a function similar to the one that we found in Equation (2.7). In particular, examples of activation functions include the linear-threshold (or ReLU) function and the *sigmoid function*[2].

In order to describe the synaptic input current $I_{s,i}(t)$, let us now assume that the postsynaptic neuron i receives M synaptic inputs from M presynaptic neurons, labeled by $j = 1, \ldots, M$. If an action potential arrives at input j at time 0, we can write the synaptic current generated in the soma of the postsynaptic neuron at time t as $w_{ij}\kappa(t)$, where w_{ij} is the synaptic weight and $\kappa(s)$ is called *synaptic kernel*, which in particular describes the temporal evolution of the synaptic current in response to a presynaptic current. In particular the synaptic weights w_{ij} will take positive values for excitatory synapses, and negative values for inhibitory synapses.

Under the assumption that the effect of the spikes sum linearly at each synapse, the total synaptic current from input j at time t_k can be formulated as

$$w_{ij} \sum_{t_k \leq t} \kappa(t - t_k) = w_{ij} \int_{-\infty}^{t} \kappa(t - \tau)\rho_j(\tau)d\tau \tag{2.11}$$

where $\rho_j(\tau)$ denotes the neural response of the presynaptic neuron j. The total synaptic current coming from all presynaptic neurons can be computed by summing over the presynaptic neurons j

$$I_{s,i} = \sum_{j=1}^{M} w_{ij} \int_{-\infty}^{t} \kappa(t - \tau)\rho_j(\tau)d\tau. \tag{2.12}$$

In order to obtain the firing-rate model, the neural response of the presynaptic neuron $\rho_j(t)$ is replaced by the firing rate of the presynaptic neuron $r_j(t)$.

[2] A sigmoid function is a mathematical function having a characteristic "S"-shaped curve or "sigmoid" curve. This function is bounded, differentiable, real and defined for all real input values; it has a non-negative derivative at each point and exactly one inflection point.

This step is typically justified by invoking mean-field arguments, which ultimately boils down to the application of the central limit theorem [7]. Under this assumption, Equation (2.12) becomes

$$I_{s,i} = \sum_{j=1}^{M} w_{ij} \int_{-\infty}^{t} \kappa(t-\tau) r_j(\tau) d\tau. \tag{2.13}$$

The synaptic kernel is usually taken to be an exponential, such as $\kappa(t) = \exp\{(-t/\tau_s)/\tau_s\}$, so that we can rewrite (2.13) as follows:

$$I_{s,i} = \sum_{j=1}^{M} w_{ij} \int_{-\infty}^{t} \frac{1}{\tau_s} e^{-\frac{(t-\tau)}{\tau_s}} r_j(\tau) d\tau. \tag{2.14}$$

In this case, Equation (2.14) can be seen as the forced response of the following dynamical model of Equation (2.15):

$$\frac{dI_{s,i}}{dt} = -\frac{1}{\tau_s} I_{s,i} + \sum_{j=1}^{M} \left(\frac{w_{ij}}{\tau_s} \right) r_j, \tag{2.15}$$

which can be rewritten as

$$\tau_s \frac{dI_{s,i}}{dt} = -I_{s,i} + \sum_{j=1}^{M} w_{ij} r_j \tag{2.16}$$

Together, (2.10) and (2.16) can be used to describe the dynamics of a population of neurons. To reduce the analysis to a single rate equation, it is often assumed that the rate dynamics is significantly slower than the synaptic current dynamics, namely $\tau_r \gg \tau_s$. Under this assumption, from (2.16), it follows that $I_{s,i} = \sum_{j=1}^{M} w_{ij} r_j$. By substituting the latter expression in (2.10), we obtain the rate dynamics

$$\tau_r \frac{dr_i}{dt} = -r_i + f\left(\sum_{j=1}^{M} w_{ij} r_j + I_{e,i} \right). \tag{2.17}$$

By introducing the matrix of synaptic weights $\boldsymbol{W} = [w_{ij}]_{i,j=1,\dots,M} \in \mathbb{R}^{M \times M}$,the vector of firing rates $\boldsymbol{r} = [r_1 \cdots r_M]^\top$, and the vector of external currents $\boldsymbol{I}_e = [I_{e,1} \cdots I_{e,M}]^\top$, (2.17) can be written in compact form as

$$\tau_r \frac{d\boldsymbol{r}}{dt} = -\boldsymbol{r} + f\left(\boldsymbol{W}\boldsymbol{r} + \boldsymbol{I}_e \right), \tag{2.18}$$

where in this case the function f is applied elementwise to the components of the vector $\boldsymbol{W}\boldsymbol{r} + \boldsymbol{I}_e$.

2.2.2 Linearized model and Dale's law

It is often relevant to consider the behavior of the firing rate around an operating point of the neural activity, that (2.18) admits an equilibrium $\bar{\boldsymbol{r}}$ obtained for a constant input current $\bar{\boldsymbol{I}}_e$ (where we make the implicit assumption that such an equilibrium does exist). To this aim, one can consider the linearization of (2.18) around such an operating point, which leads to the linear model:

$$\frac{d\boldsymbol{x}}{dt} = \frac{1}{\tau_r}\left(-\boldsymbol{I} + \alpha \boldsymbol{W}\right)\boldsymbol{x} + \alpha \boldsymbol{u}, \tag{2.19}$$

where $\boldsymbol{I}$ denotes the identity matrix, $\boldsymbol{x} = \boldsymbol{r} - \bar{\boldsymbol{r}}$, $\boldsymbol{u} = \boldsymbol{I}_e - \bar{\boldsymbol{I}}_e$, and $\alpha = \frac{d}{dx} f(x)\Big|_{x=\boldsymbol{W}\bar{\boldsymbol{r}}+\bar{\boldsymbol{I}}_e} > 0$[3].

Neurons are usually classified as either *excitatory* or *inhibitory*, depending on the effects they have on all of their post-synaptic nodes. This particular behaviour of neurons is summarized by Dale's principles, which basically states that that a neuron performs the same chemical action at all of its synaptic connections to other cells, regardless of the type of the target cell.

To put it simply, the neuron cannot excite some of its post-synaptic neurons and inhibit others. As a consequence, the entries of any given column of the synaptic matrix $\boldsymbol{W}$, must have the same sign, either positive or negative.

By partitioning the vector $\boldsymbol{x}$ in (2.19) into excitatory and inhibitory parts

$$\boldsymbol{x} = \left[\begin{array}{c} \boldsymbol{x}_e \\ \hline \boldsymbol{x}_i \end{array}\right],$$

where $\boldsymbol{x}_e$ and $\boldsymbol{x}_i$ comprise the excitatory and inhibitory components, respectively, and the synaptic matrix $\boldsymbol{W}$ and input vector $\boldsymbol{u}$ accordingly

$$\boldsymbol{W} = \left[\begin{array}{c|c} \boldsymbol{W}_{ee} & -\boldsymbol{W}_{ei} \\ \hline \boldsymbol{W}_{ie} & -\boldsymbol{W}_{ii} \end{array}\right], \quad \boldsymbol{u} = \left[\begin{array}{c} \boldsymbol{u}_e \\ \hline \boldsymbol{u}_i \end{array}\right]$$

where all blocks $\boldsymbol{W}_{ee}$, $\boldsymbol{W}_{ei}$, $\boldsymbol{W}_{ie}$, $\boldsymbol{W}_{ii}$ have non-negative entries, we can

[3] The coefficient α is assumed to be positive since commonly used activation functions are monotonically increasing.

rewrite (2.19) as follows

$$\begin{cases} \dfrac{d\boldsymbol{x}_e}{dt} = \left(-\dfrac{1}{\tau_r}\boldsymbol{I} + \alpha\boldsymbol{W}_{ee}\right)\boldsymbol{x}_e - \alpha\boldsymbol{W}_{ei}\boldsymbol{x}_i + \alpha\boldsymbol{u}_e \\ \dfrac{d\boldsymbol{x}_i}{dt} = \left(-\dfrac{1}{\tau_r}\boldsymbol{I} - \alpha\boldsymbol{W}_{ii}\right)\boldsymbol{x}_i + \alpha\boldsymbol{W}_{ie}\boldsymbol{x}_e + \alpha\boldsymbol{u}_i \end{cases} \tag{2.20}$$

Having mixed excitatory-inhibitory subpopulations leads to models that exhibit richer dynamics than single population models. In fact, in models with excitatory and inhibitory subpopulations, the full synaptic weight matrix is not symmetric and network oscillations can arise.

The model represented by equations (2.20) will be exploited to investigate and eventually describe the behavior of interacting populations of inhibitory and excitatory neurons in the brain.

2.3 Excitatory-Inhibitory Model

As we discussed in the previous chapters, brain functioning depends on the interaction among different regions of the cortex, which exchange information via a complex connectivity network and work together in a coordinated manner to realize cognitive tasks.

In particular, it is known that the nervous tissue can generate oscillatory activity in many ways, driven either by the mechanism within the single neurons or by the interaction between different clusters of neurons.
As for single neurons, oscillations can arise either as oscillations in membrane potential or as rhythmic patterns of action potentials and spikes, which consequently produce oscillatory activation on the post-synaptic neurons. On the other hand, at the level of neural populations and networks, synchronized activity of large numbers of neurons can produce macroscopic oscillations, which can be measured by techniques such as EEG (i.e. "electroencephalography").

The interaction between neurons usually cause oscillations at frequencies that might differ from the firing frequency of individual neurons and a well-known example of macroscopic neural oscillations is so-called *alpha rhythm*, i.e. neural oscillations in the frequency range of 8–12 Hz. Moreover, neural oscillations also play an important role in many neurological disorders and

are typically characterized by wide-ranging functions that vary for different types of oscillatory activity.

In this section, we continue with the analysis by presenting a simplified yet tractable neural model based on the interaction between excitatory and inhibitory neurons that will be discussed and then exploited throughout the text to study the oscillatory phenomena.

More specifically, we will start by analytically investigating the activity of a simple neuronal model and then we will attempt to show how the model derived from it can be actually useful to describe and disclose the dynamics of the underlying neural coding and communication framework.

2.3.1 A simplified E-I model

Let us now consider a simple neural assembly representing a particular neural population $\mathscr{P}$, shown in Figure 2.4 and comprising two distinct yet interacting subgroups $\mathcal{S}_e$ and $\mathcal{S}_i$ of interconnected neurons, such that the synapses of the former are excitatory while the synapses of the latter are inhibitory: this means that they will respectively increase or reduce the likelihood that the neuron will spike an action potential, as we discussed in the previous chapters.

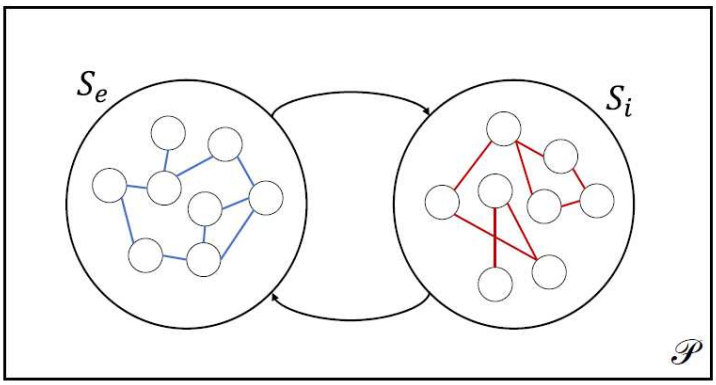


Figure 2.4: Schematic representation of the neural assembly we take into account: in particular, the neural population $\mathscr{P}$ is divided into two distinct subgroups of neurons, respectively $\mathcal{S}_e$ and $\mathcal{S}_i$, according to the type of their synapses.

We let the (linearized) firing rate dynamics of such neural population be described by (2.20). To further reduce the complexity of this model, we assume that each subgroup $\mathcal{S}_k$ (with $k = e, i$) can be described by its mean activity x_k, meaning that we average the neural activity of each subgroup $\mathcal{S}_k$ of the population $\mathcal{P}$, in order to consider separately the two subgroups. Each subgroup $\mathcal{S}_k$ of the population $\mathcal{P}$ interacts with the other one $\mathcal{S}_j$ (with $j = i, e$) through a link of weight ω_{jk} (i.e. from k to j) and with themselves with a recurrent link of weight γ_k; we also assume that both the subgroups $\mathcal{S}_e$ and $\mathcal{S}_i$ are susceptible to receive an input, respectively u_e and u_i, which may represent an external current impinging on the corresponding subgroup. The input value u_k might also be affected by a bias β_k, through which we can describe a possible input conditioning.

A schematic representation of the system we are considering is illustrated in Figure 2.6. We can now describe the dynamics of the mean activity x_e and

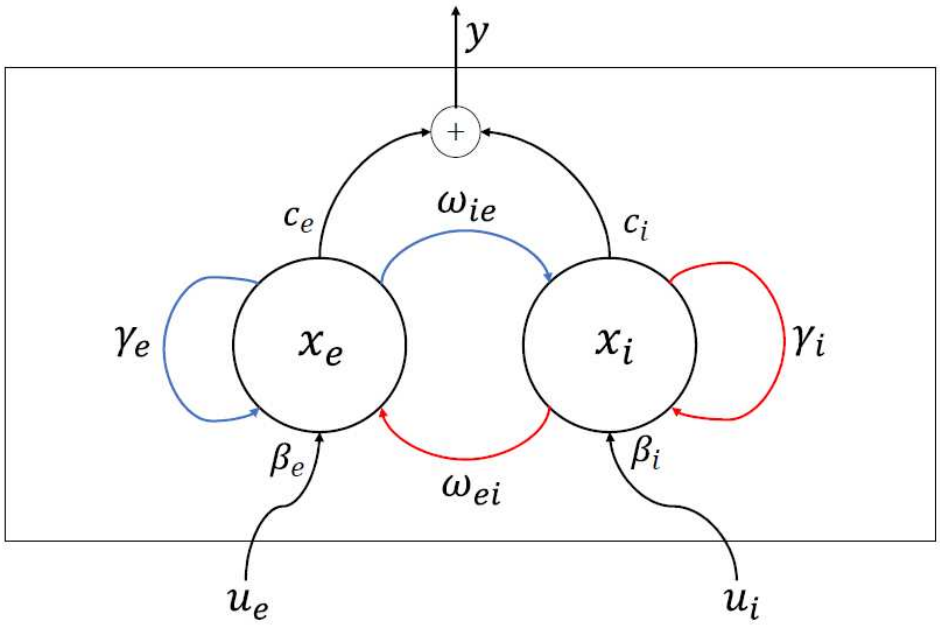


Figure 2.5: This is a schematic representation of a single neural population $\mathcal{P}$, where the two excitatory and inhibitory subgroups $\mathcal{S}_e$ and $\mathcal{S}_i$ are described through their mean activity, respectively $x_e(t)$ and $x_i(t)$.

x_i of each subgroup with the following set of differential equations:

$$\begin{cases} \dot{x}_e = \gamma_e x_e - \omega_{ei} x_i + \beta_e u_e \\ \dot{x}_i = \omega_{ie} x_e + \gamma_i x_i + \beta_i u_i. \end{cases} \tag{2.21}$$

We assume that the recurrent weight γ_e of the excitatory subgroup $\mathcal{S}_e$ is equal to the recurrent weight γ_i of the inhibitory subgroup $\mathcal{S}_i$, so that $\gamma_e = \gamma_i = -\gamma$, with $\gamma \in \mathbb{R}^+$. We make the simplifying assumption that the links between the subgroups $\mathcal{S}_e$ and $\mathcal{S}_i$ have the same weight ω, with $\omega \in \mathbb{R}_{\geq 0}$: this means that we can write $\omega_{ei} = -\omega \leq 0$ and consequently $\omega_{ie} = \omega \geq 0$.

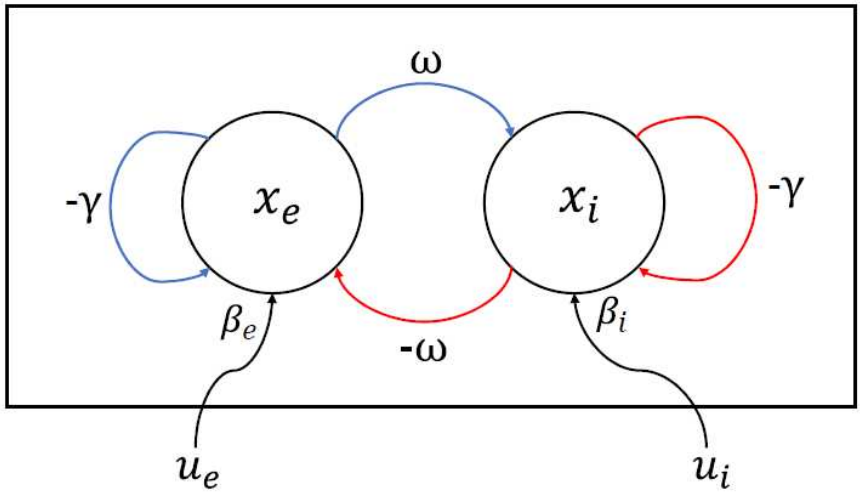


Figure 2.6: This is a schematic representation of the simplified model, with $\gamma_e = \gamma_i = -\gamma$ and with $\omega_{ie} = -\omega_{ei} = \omega$. In particular, the dynamics of the model is described by the set of differential equations shown in Equations (2.21).

Please note that the weight ω_{ei} will only assume non-positive values, since it represents the interaction of the inhibitory subgroup $\mathcal{S}_i$ on the excitatory one $\mathcal{S}_e$, meaning that it will tend to reduce the activity of this last one. On the other hand, the weight ω_{ie} will consequently assume only non-negative values.

At a first analysis, all the variables will be considered time-invariant, meaning that they are not function of the time.

It is possible to rewrite the set of equations in a matrix form and obtain the following state-space representation of our LTI system:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} = \begin{bmatrix} -\gamma & -\omega \\ \omega & -\gamma \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \beta_e \\ \beta_i \end{bmatrix} \boldsymbol{u}, \tag{2.22}$$

where $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$ is the state matrix, $\boldsymbol{B} \in \mathbb{R}^2$ is the input matrix, $\boldsymbol{x} = \begin{bmatrix} x_e & x_i \end{bmatrix}^T$ is the state vector, and $\boldsymbol{u} = \begin{bmatrix} u_e & u_i \end{bmatrix}^T$ is the input of the model, namely the vector that contains the signals to be transmitted from the particular node through the network.

It is worth noting that the state matrix $\boldsymbol{A}$ can be written as

$$\boldsymbol{A} = -\gamma \boldsymbol{I} + \boldsymbol{W}, \quad \boldsymbol{W} = \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix}.$$

which is a simplified version of the model (2.20) for scalar excitatory and inhibitory subpopulations.

In order to achieve our final model, we will need to make another assumption on the possible interconnections between different populations $\{\mathscr{P}_k\}$, where $k = 1, \dots, N$, each of them comprising an excitatory subgroup $\mathscr{S}_{e,k}$ and an inhibitory subgroup $\mathscr{S}_{i,k}$: in fact, it is reasonable to assume that each population $\mathscr{P}_k$ can communicate and transmit information with another similar population $\mathscr{P}_j$, with $j \neq k$, only via excitatory connections and not via the inhibitory ones, but this will be detailed later in the section.

At first, let us assume that the output signal is a linear combination of the state of the excitatory and inhibitory subgroups, respectively x_e and x_i. This yields to the following output Equation (2.23):

$$\begin{aligned} y(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}u(t) \\ &= \begin{bmatrix} c_e & c_i \end{bmatrix} \boldsymbol{x}(t) \end{aligned} \tag{2.23}$$

where $\boldsymbol{C} \in \mathbb{R}^{1\times 2}$ is the output matrix that characterizes the output from the state vector $\boldsymbol{x}$ and $\boldsymbol{D} \in \mathbb{R}^{2\times 1}$ is the feedforward matrix. In this case, we assume that $\mathbf{D} = \boldsymbol{O}_{2\times 1}$, so that we have no feedforward of the input vector.

Finally, the model of the Excitatory-Inhibitory (E-I) system Σ reads as:

$$\begin{cases} \dot{\boldsymbol{x}} = \begin{bmatrix} -\gamma & -\omega \\ \omega & -\gamma \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \beta_e \\ \beta_i \end{bmatrix} \boldsymbol{u} \\ \\ y = \begin{bmatrix} c_e & c_i \end{bmatrix} \boldsymbol{x}. \end{cases} \tag{2.24}$$

In order to determine the eigenvalues of the state matrix $\boldsymbol{A}$ we simply need to compute the solutions of the following characteristic equation:

$$\begin{aligned}\det(\lambda \boldsymbol{I} - \boldsymbol{A}) &= \det \begin{bmatrix} \lambda + \gamma & \omega \\ -\omega & \lambda + \gamma \end{bmatrix} \\ &= (\lambda + \gamma)^2 + \omega^2 \\ &= \lambda^2 + 2\lambda\gamma + \gamma^2 + \omega^2 = 0\end{aligned} \tag{2.25}$$

The two solutions $\lambda_{1,2} \in \mathbb{C}$ of Equation (2.25) are

$$\lambda_{1,2} = -\gamma \pm i\omega, \tag{2.26}$$

therefore the system has a pair of complex conjugate eigenvalues, meaning that the model can exhibit stable or unstable oscillations, according to the value of the parameter $-\gamma$, which is the real part of the eigenvalues $\lambda_{1,2}$. In particular, our model Σ will show asymptotic stability and the oscillations will asymptotically converge to a stable point if the value of γ is strictly positive, while the model will show instability and the oscillations will exponentially diverge to infinity with a strictly negative γ.

If $\gamma = 0$ we obtain two imaginary solutions $\lambda_{1,2} = \pm\omega$ of the characteristic Equation (2.25) and our model Σ will then exhibit simple stability, thus stable oscillations. It is clear that we are interested in the case where γ is chosen to be strictly positive, since we aim to obtain a stable oscillating model.

Let us now consider the transfer function $H(s)$ of the LTI system Σ shown in equations (6.1), which can be computed as follows:

$$\begin{aligned}H(s) &= \boldsymbol{C}\big(s\boldsymbol{I} - \boldsymbol{A}\big)^{-1}\boldsymbol{B} = \\ &= \begin{bmatrix} c_e & c_i \end{bmatrix} \begin{bmatrix} s + \gamma & \omega \\ -\omega & s + \gamma \end{bmatrix}^{-1} \begin{bmatrix} \beta_e \\ \beta_i \end{bmatrix} = \\ &= \begin{bmatrix} c_e & c_i \end{bmatrix} \frac{1}{(s+\gamma)^2 + \omega^2} \begin{bmatrix} s + \gamma & -\omega \\ \omega & s + \gamma \end{bmatrix} \begin{bmatrix} \beta_e \\ \beta_i \end{bmatrix} = \\ &= \frac{1}{(s+\gamma)^2 + \omega^2} \begin{bmatrix} c_e & c_i \end{bmatrix} \begin{bmatrix} \beta_e(s+\gamma) - \beta_i\omega \\ \beta_e\omega + (s+\gamma)\beta_i \end{bmatrix} = \\ &= \frac{[\beta_e c_e + \beta_i c_i](s+\gamma) + [\beta_e c_i - \beta_i c_e]\omega}{(s+\gamma)^2 + \omega^2}.\end{aligned} \tag{2.27}$$

If we reasonably assume that for every neural population $\mathscr{P}$ we are considering only the excitatory subgroup $\mathcal{S}_e$ can receive an input signal u_e and

consequently transmit an output signal y to another excitatory subgroup of a different neural population, we can simplify our model Σ and rewrite the transfer function $H(s)$ as:

$$H(s) = K\frac{s+\gamma}{(s+\gamma)^2+\omega^2}, \tag{2.28}$$

where $\boldsymbol{B} = \beta\begin{bmatrix}1 & 0\end{bmatrix}^\top$, $\boldsymbol{C} = c\begin{bmatrix}1 & 0\end{bmatrix}$ and $K := \beta c$ is the gain of the transfer function.

Please note that, even though this might seem a strong approximation, it appears significantly reasonable, since in neural networks the information is normally conveyed via excitatory synapses, while the inhibitory synapses typically serve to controlling and thus limiting the overall communication when needed.

2.3.2 Frequency response analysis

We will now restrict the analysis of the transfer function $H(s)$ shown in Equation (2.28) to the particular case where the input signal of the system Σ is taken to be a sinusoidal signal with amplitude u_0 and varying frequency σ

$$u(t) = u_0 e^{j\sigma t}.$$

In this case, we will obtain the frequency response $H(j\sigma)$ of the system Σ in response to the sinusoidal input $u(t)$, i.e.

$$H(j\sigma) = K\frac{\gamma + j\sigma}{(j\sigma + \gamma)^2 + \omega^2} = K\frac{\gamma + j\sigma}{(\gamma^2 + \omega^2 - \sigma^2) + j2\gamma\sigma}. \tag{2.29}$$

Under the assumption that the input signal is sinusoidal, the output $y(t)$ of the system Σ can be easily found exploiting the frequency response $H(j\sigma)$ shown in Equation (4.2), which yields to

$$y(t) = A|H(j\sigma)| \cdot e^{j\sigma t + \arg(H(j\sigma))} = y_0 e^{j\sigma t + \varphi}. \tag{2.30}$$

According to Equation (2.30), the output signal $y(t)$ will as well be sinusoidal with amplitude $y_0 = |H(j\sigma)|u_0$ and shifted by a phase angle equal to $\arg(y) = \arg(H(j\sigma))$ with respect to the input signal $u(t)$.

Amplitude analysis

First of all, we are interested in finding whether the transfer function $H(j\sigma)$ shown in equation 4.2 has a resonance peak and for which values of the parameters γ and ω, in order to understand at which frequency $\sigma(\gamma, \omega)$ the system Σ will give the maximum amplification of the input signal u.

Lemma 2.3.1. *Given the following frequency response*

$$H(j\sigma) = K\frac{\gamma + j\sigma}{(\gamma^2 + \omega^2 - \sigma^2) + j2\gamma\sigma}$$

with γ and ω strictly positive, the frequency response has a resonance frequency σ_{pk} and thus a resonance peak $|H(j\sigma_{pk})|$, with $\sigma_{pk} \neq 0$, if and only if

$$-\gamma > \left(\sqrt{2 + \sqrt{5}}\right)\omega.$$

If the resonance condition holds true, the resonance frequency σ_{pk} is located exactly at

$$\sigma_{pk} = \sqrt{-\gamma^2 + \omega\sqrt{\omega^2 + 4\gamma^2}}.$$

Proof. Let us start by analytically computing the module of the frequency response $H(j\sigma)$:

$$|H(j\sigma)| = \left(\frac{\sigma^2 + \gamma^2}{(\gamma^2 + \omega^2 - \sigma^2)^2 + 4\gamma^2\sigma^2}\right)^{\frac{1}{2}}. \tag{2.31}$$

In order to find the resonance frequency σ_{pk}, we can compute the derivative of $|H(j\sigma)|$ with respect to σ and then set it equal to zero. Since the square root is not relevant for the computation of the derivative, we can compute directly the derivative of $|H(j\sigma)|^2$

$$\frac{d}{d\sigma}\left(|H(j\sigma)|^2\right) = \frac{d}{d\sigma}\left[\frac{\sigma^2 + \gamma^2}{(\gamma^2 + \omega^2 - \sigma^2)^2 + 4\gamma^2\sigma^2}\right] = \frac{d}{d\sigma}\left[\frac{n(\sigma)}{d(\sigma)}\right] = 0,$$

where $n(\sigma) = \sigma^2 + \gamma^2$ and $d(\sigma) = (\gamma^2 + \omega^2 - \sigma^2)^2 + 4\gamma^2\sigma^2$.

The last equality can be rewritten as

$$\frac{d}{d\sigma}\left[\frac{n(\sigma)}{d(\sigma)}\right] = \frac{n'(\sigma)d(\sigma) - n(\sigma)d'(\sigma)}{d(\sigma)^2} = 0,$$

so the analysis ultimately reduces to computing when numerator is equal to zero,

$$n'(\sigma)d(\sigma) - n(\sigma)d'(\sigma) = 0.$$

Thus, we can write:

$$2\sigma\left[(\gamma^2 + \omega^2 - \sigma^2)^2 + 4\sigma^2\gamma^2\right] - (\gamma^2 + \sigma^2)\left[8\gamma^2\sigma - 4\sigma(\gamma^2 + \omega^2 - \sigma^2)\right] = 0,$$

we then divide by 2σ, by assuming that $\sigma \neq 0$, which leads to

$$\begin{aligned}\gamma^4 + \omega^4 + \sigma^4 + 2\gamma^2\omega^2 - 2\gamma^2\sigma^2 - 2\omega^2\sigma^2 + 4\gamma^2\sigma^2 \\ - \left[2\gamma^4 - 2\omega^2\gamma^2 + 2\gamma^2\sigma^2 - 2\omega^2\sigma^2 + 2\sigma^4\right] = 0\end{aligned}$$

which, after some trivial computational steps, can be reduced to the following biquadratic equation:

$$\sigma^4 + 2\gamma^2\sigma^2 + \gamma^4 - \omega^4 - 4\omega^2\gamma^2 = 0. \tag{2.32}$$

Let us now make the following substitution $z = \sigma^2$, with $z > 0$, which consequently yields to the following quadratic equation

$$z^2 + 2\gamma^2 z + \gamma^4 - \omega^4 - 4\omega^2\gamma^2 = 0.$$

The discriminant of the equation is always strictly positive

$$\frac{\Delta}{4} = \gamma^4 + \omega^4 - \gamma^4 + 4\gamma^2\omega^2 = \omega^2(\omega^2 + 4\gamma^2),$$

and the solutions of the quadratic equation above are

$$z_{1,2} = -\gamma^2 \pm \sqrt{\frac{\Delta}{4}} = -\gamma^2 \pm \sqrt{\omega^2(\omega^2 + 4\gamma^2)}.$$

Only the solution with the plus symbol is acceptable, since z must be strictly positive by assumption. By recalling the substitution $z = \sigma^2$, we can write

$$\sigma^2 = -\gamma^2 \pm \sqrt{\omega^2(\omega^2 + 4\gamma^2)},$$

$$\sigma_{1,2} = \pm\sqrt{-\gamma^2 \pm \sqrt{\omega^2(\omega^2 + 4\gamma^2)}}.$$

However, since we are dealing with positive frequency, only the solution with the plus symbol is acceptable and thus we obtain a unique solution for the peak frequency σ_{pk} (except for the trivial solution $\sigma = 0$), which finally is

$$\sigma_{pk} = \sqrt{-\gamma^2 \pm \sqrt{\omega^2(\omega^2 + 4\gamma^2)}}.$$

This last expression is valid only when the argument of the square root is positive, so we also need to understand for which values of γ and ω this is true. Let us set $\varepsilon = \frac{\gamma^2}{\omega^2}$, we can rewrite the argument under the square root as a function of ε, leading us to

$$\begin{aligned} & -\gamma^2 + \omega^2\sqrt{1 + 4\tfrac{\gamma^2}{\omega^2}} > 0 \\ \Leftrightarrow \quad & \sqrt{1 + 4\varepsilon} > \varepsilon \\ \Leftrightarrow \quad & \varepsilon^2 - 4\varepsilon - 1 < 0 \end{aligned}$$

and it is particularly easy to verify that the last inequality holds true if and only if $\varepsilon \in (0; 2 + \sqrt{5}]$, or alternatively when

$$0 < \gamma < \left(\sqrt{2 + \sqrt{5}}\right)\omega. \tag{2.33}$$

□

Remark. Please notice that the resonance condition above can be further approximated to $0 < \gamma < \left(\sqrt{2+\sqrt{5}}\right)\omega \approx 2\omega$. When this particular condition does not hold true, the frequency response $|H(j\sigma)|$ will not have a resonance peak, but we showed that it is reasonable to keep the parameter γ quite small compared to ω, since we are working with an oscillator model: in fact, the lower γ is and the more the model will show an oscillating behaviour.

In particular, when we consider γ to be much less than ω, thus the ratio ρ between the parameter γ and the parameter ω, defined as

$$\rho = \frac{\gamma}{\omega},$$

tends to zero, the resonance condition $0 < \gamma < 2\omega$ always holds true and σ_{pk} can be approximated with ω,

$$\sigma_{pk} = \sqrt{-\gamma^2 \pm \sqrt{\omega^2(\omega^2 + 4\gamma^2)}} \approx \omega.$$

This means that, whenever we take ρ to be near to zero, it is possible to trigger the system Σ with a sinusoidal signal $u = u_0 \sin(\omega_0 t)$ and obtain the maximum amplification of the signal when the parameter ω is very near to the frequency of the incoming signal, i.e. ω_0. We will say that the system has the be *tuned* on the same frequency of the input signal, in order to give rise to the maximum amplification possible. As we will see, the entity of the amplification will not only be ruled by the choice of the parameters ω and γ, but also by the choice of the parameters β and c, which represent the signal conditioning on the input signal and on the output signal, respectively.

In Figure 2.7 we show how the bode plots change according to the entity of γ with respect to ω which is fixed and set to be equal to the incoming signal frequency ω_0.

We might also be interested in evaluating the entity of the amplification, under the assumption that ρ tends to zero. By substituting $\sigma = \sigma_{pk} \approx \omega$ in (2.31) we obtain

$$\begin{aligned} |H(j\sigma_{pk})| \approx |H(j\omega)| &= K\left(\frac{\omega^2+\gamma^2}{\gamma^2+\omega^2-\omega^2+4\gamma^2\omega^2}\right)^{\frac{1}{2}} \\ &= K\left(\frac{\omega^2+\gamma^2}{\gamma^2+4\gamma^2\omega^2}\right)^{\frac{1}{2}}, \end{aligned} \tag{2.34}$$

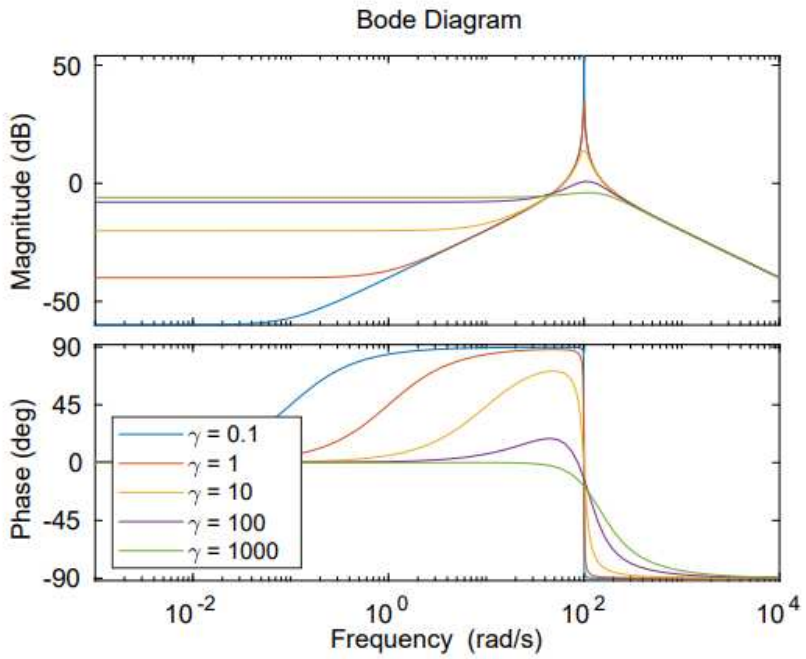


Figure 2.7: Bode plots of the frequency response $H(j\omega)$ for different values of γ and fixed ω. In particular, the natural frequency of the oscillator is fixed to $\omega_0 = 100$ rad/s, while $b = c = 10$. As we can see, as the ratio $\rho = \gamma/\omega$ gets smaller, the resonance peak gets narrower and the amplification given by the resonance increases.

which can be further simplified into (recall that we are assuming $\rho \to 0$)

$$|H(j\omega)| = K\frac{1}{2\gamma}. \tag{2.35}$$

In particular, we will obtain an amplification of the input signal whenever the frequency response $H(j\sigma)$ shows a resonance peak and its module $|H(j\sigma)|$ is greater than 1, thus

$$|H(j\omega)| = K\frac{1}{2\gamma} > 1 \quad \iff \quad K = \beta c > 2\gamma. \tag{2.36}$$

This last condition seems to be quite legit under the assumption that ρ tends to zero, in fact if we divide both the terms by the quantity ω we obtain

$$\frac{\beta c}{\omega} > \frac{2\gamma}{\omega} = 2\rho \approx 0, \tag{2.37}$$

meaning that we will just need that β and c are strictly positive quantities in order to have an amplification. This is due to the fact that keeping γ very low with respect to ω lets us have a highly-responsive system around the peak frequency.

Phase Analysis

Lemma 2.3.2. *Given the following frequency response*

$$H(j\sigma) = K\frac{\gamma + j\sigma}{(\gamma^2 + \omega^2 - \sigma^2) + j2\gamma\sigma},$$

the phase of $H(j\sigma)$ is zero for

$$\sigma = \sigma^* = \sqrt{\omega^2 - \gamma^2}.$$

Proof. Let us start by computing the phase[4] of the frequency response $H(j\sigma)$,

$$\begin{aligned}\arg\left(H(j\sigma)\right) &= \arg(K) + \arg(\gamma + j\sigma) - \arg\left((\gamma^2 + \omega^2 - \sigma^2) + j2\gamma\sigma\right) = \\ &= \arctan\left(\frac{\sigma}{\gamma}\right) - \arctan\left(\frac{2\gamma\sigma}{\gamma^2 + \omega^2 - \sigma^2}\right).\end{aligned}$$

We can now set the phase to zero in order to find the value σ^*

$$\arg(H(j\sigma^*)) = 0 \quad \Leftrightarrow \quad \arctan\left(\frac{\sigma^*}{\gamma}\right) = \arctan\left(\frac{2\gamma\sigma^*}{\gamma^2 + \omega^2 - \sigma^{*2}}\right).$$

This holds true if and only if

$$\frac{\sigma^*}{\gamma} = \frac{2\gamma\sigma^*}{\gamma^2 + \omega^2 - \sigma^{*2}}$$

and for $\sigma^* \neq 0$, $\gamma \neq 0$ and $\gamma^2 + \omega^2 - \sigma^{*2} \neq 0$ we can write

$$\gamma^2 + \omega^2 - \sigma^{*2} = 2\gamma^2$$

[4]For a generic complex number $z = a + jb$, we recall that the phase can be computed as $\arg(z) = \arctan(b/a)$.

which ultimately leads to

$$\sigma^* = \sqrt{\omega^2 - \gamma^2}.$$

□

Remark. Please note that, even in this case, when γ is taken to be much smaller than ω, we can approximate σ^* with the parameter ω. If the frequency response $H(j\sigma)$ has phase approximately equal to zero when σ is tuned to ω, the output signal phase $\arg(y)$, as described in Equation (2.30), will then be approximately equal to zero. This means that our simplified model Σ can tune itself on the input signal frequency in order to amplify it without introducing any delay into the output signal, since the output signal y shows the same phase as the input signal u.

In the sequel of the chapter, we will refer to a particular communication framework as shown in the block diagram of Figure 3.1, where a white noise signal $n(t)$ is added to the system output $y(t)$ in order to take into account the fact that the communication between different neuron regions, each of them described by our model Σ, might be disturbed by a background noise.

Part II

Interplay between Communication and Synchronization

Chapter 3

Problem formulation

3.1 Communication framework

From now on, we will be considering the communication framework illustrated in Figure 3.1 where $u(t)$ is the input signal, Σ_{tot} is a generic network of N oscillators $\Sigma_1, \dots, \Sigma_N$ with transfer function $H_{tot}(s, \{\omega_{n,i}\}_{i=1}^N)$, which is itself function of the natural frequencies $\omega_{n,i}$ of the single oscillators, $y(t)$ is the output signal, $n(t)$ is an additive time-varying noise, which represents the noisiness during the transmission of the signal. Lastly, $\tilde{y}(t) = y(t) + n(t)$ is the corrupted output signal which contains the information to be transmitted $y(t)$ and the noise $n(t)$.

In this text, $n(t)$ is considered to be a white noise, meaning that it is a random signal with equal intensity at different frequencies. In particular, $n(t)$ will have zero mean and constant variance σ^2. Moreover, the input $u(t)$ is considered to be sinusoidal with constant amplitude and frequency. This choice is justified by the fact that the frequency content of the sensory inputs of the brain are often concentrated around a single frequency.

This particular framework will be useful to investigate either how the different interconnections between different systems affect the communication and how the natural frequencies $\{\omega_{n,1}, \dots, \omega_{n,N}\}$ of the single systems can be changed, adapted, or *tuned*, in order to obtain the best transmission of information possible.

In order to do so, we will start by studying a single system $\Sigma_1(\omega_{n,1})$ and then we will study what happens if we add another system $\Sigma_2(\omega_{n,2})$ interconnected

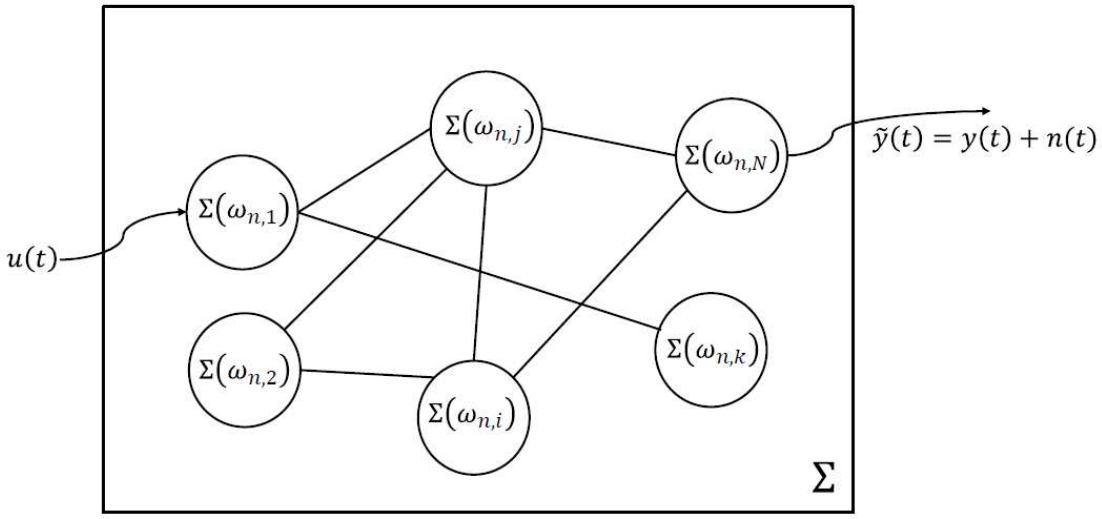


Figure 3.1: Schematic of the communication framework considered in the sequel of the text. The noisiness of the channel is described through an additive white noise signal $n(t)$ which corrupts the output signal $y(t)$ into the corrupted signal $\tilde{y}(t)$. In particular, Σ is taken to be a generic network of N different oscillators, where we can define an input node, i.e. the input signal receiver, and an output node, i.e. the output signal transmitter.

to the first one: some simpler cases of interconnection will be presented and analyzed, like the series or the feedback of two systems $\Sigma_1(\omega_{n,1})$ and $\Sigma_2(\omega_{n,2})$.

This appears to be a problem of optimization of the single natural frequencies: the aim of the next chapters will be based on finding which set of natural frequencies $\{\omega_{n,1}, \ldots, \omega_{n,N}\}$ optimize the communication and the transmission of the information, given the overall transfer function $H_{tot}(s; \{\omega_{n,i}\}_{i=1}^{N})$.

One of the simplest, yet reasonable, metric we could exploit in order to thoroughly formalize the problem of the optimization of the natural frequencies is the *signal-to-noise ratio*, i.e., the ratio between the power of the meaningful signal, the information to be transmitted, and the power of the additive noise signal, which corrupts and affects the overall communication. This specific metric focuses on the propagation of the signal over the background noise, meaning that it will characterize how *powerful* is the signal to be transmitted with respect to the disturbing noise. Please notice that the metric to

be chosen for the optimization problem is arbitrary and thus not constraining: other metrics could be used to tackle the problem we presented, yet the signal-to-noise ratio appears to be quite reasonable and appropriate for the communication framework we are taking into consideration, at least in a preliminary stage.

The signal-to-noise ratio will be then presented and extensively described in the next chapter.

3.2 Signal-to-noise Ratio

In this section, we introduce a specific metric which can be used to describe the effective propagation of a signal in a noisy communication framework as the one illustrated in the block diagram of Figure 3.1: the *signal-to-noise ratio*, namely the ratio between the signal power and the background noise power. The signal-to-noise ration is often expressed in decibels.

Definition 3.2.1 (Signal-to-noise ratio)**.** The signal-to-noise ratio (or SNR) is defined as the ratio between the power of a signal (meaningful input) and the power of the background noise (meaningless input):

$$SNR = \frac{P_{signal}}{P_{noise}}.$$

If the signal $s(t) = A\sin(\omega t)$ is taken to be a sine wave with constant amplitude A and angular frequency ω, it is possible to demonstrate that the power of the sinusoidal signal is equal to its squared Root Mean Square (RMS) value, namely

$$P_{s(t)} = A_{rms} = \frac{A^2}{2} = \left(\frac{A}{\sqrt{2}}\right)^2.$$

The (statistical) power of a white noise $n(t)$ with zero mean μ and variance σ^2 is equal to its variance, namely

$$P_n = var[n(t)] = \sigma^2.$$

The SNR can be finally rewritten as

$$SNR = \frac{A^2}{2\sigma^2} = \frac{A_{rms}^2}{\sigma^2} = \left(\frac{A_{rms}}{\sigma}\right)^2, \tag{3.1}$$

and in decibels

$$SNR_{dB} = 10\log_{10}\left(\frac{A_{rms}^2}{\sigma^2}\right) = 20\log_{10}\left(\frac{A_{rms}}{\sigma}\right). \tag{3.2}$$

As we can deduce from Equation (3.1), the signal-to-noise ratio is strictly related to the amplitude of the signal to be transmitted, meaning that the greater the amplitude of the sinusoidal signal is with respect to the noise the better the propagation of the signal over the background noise will be.

Now we can apply Equation (3.1) to the communication framework described in the preceding chapter by replacing $s(t)$ with the output signal $y(t)$.

Under the assumption that the input signal is sinusoidal with constant amplitude A and angular frequency ω_0,

$$u(t) = A\sin(\omega_0 t),$$

the output signal $y(t)$ can be computed exploiting the frequency response $|H_{tot}(j\omega)|$, which leads to

$$y(t) = A|H_{tot}(j\omega_0)|\sin(\omega_0 + \arg(|H_{tot}(j\omega_0)|)) = A'\sin(\omega_0 + \varphi),$$

where we denoted with A' the amplitude of the output signal $y(t)$ and with φ the phase of $|H_{tot}(j\omega)|$ particularly evaluated at ω_0.

Thus, the signal-to-noise ratio can be easily derived from Equation (3.1)

$$SNR = \frac{(A')^2}{2\sigma} = \frac{1}{2\sigma^2}A^2|H_{tot}(j\omega_0)|, \tag{3.3}$$

where σ and A are given parameters of the noise $n(t)$ and the input signal $u(t)$, respectively.

3.3 Optimization problem

The focus of this section is centered on defining and formalizing the optimization problem that was highlighted and pointed out in the previous section: since the output signal $y(t)$ is corrupted into $\tilde{y}(t)$ by a random additive noise signal $n(t)$, in order to have the best propagation of the output signal possible, we will reasonably need to maximize the signal-to-noise ratio. In other

words, we will need to minimize the influence of the noise signal $n(t)$ over the information-transmitting signal $y(t)$. In fact, under this circumstance, the corrupted signal $\tilde{y}(t)$ can be then approximated to the original uncorrupted signal $y(t)$.

As we can see from (3.3), the signal-to-noise ratio is directly proportional to the module of the frequency response of Σ_{tot}, that is $|H_{tot}(j\omega)|$, specifically evaluated at ω_0, since either A and σ are constant, hence fixed parameters.

As we discussed before, the frequency response $|H_{tot}(j\omega)|$ is also function of the natural frequencies of the single systems $\Sigma_1, \dots, \Sigma_N$, i.e. the set of natural frequencies $\{\omega_{n,1}, \dots, \omega_{n,N}\}$. For simplicity, we will denote the set of natural frequencies with $\{\omega_{n,i}\}$.

This also means that the optimization problem boils down to finding for which set of values of the natural frequencies $\{\omega_{n,i}\}$ we obtain the maximum value of the signal-to-noise ratio, which also means having the best propagation possible of the information-related signal over the random noise in the communication framework. Please notice that in the analysis the frequency of the input signal, that is ω_0, will be fixed and thus constant.

In mathematical terms, in order to find the optimal choice of the natural frequencies, we could define our problem in the following way:

$$\arg\max_{\{\omega_{n,i}\}} \; SNR \iff \arg\max_{\{\omega_{n,i}\}} |H_{tot}(j\omega_0; \{\omega_{n,i}\})|. \tag{3.4}$$

Another question that spontaneously arises is related to the relationship between the natural frequency and the possible synchronization between the E-I systems taken into consideration. More specifically, with the term "synchronization" we refer to the fact that two or more E-I systems might be characterized by natural frequencies that are significantly near to each other, hence approximately the same. This property seems to be quite interesting since the potential synchronization between different E-I systems might result in a better, if not optimal, propagation of signals carrying the information throughout the network.

Chapter 4

Single E-I system

Let us now suppose that Σ_{tot} comprises only one E-I system Σ_1, whose state matrix $\boldsymbol{A}_1$ is characterized by its corresponding natural frequency $\omega_{n,1}$, namely

$$\boldsymbol{A}_1 = \begin{bmatrix} -\gamma & -\omega_{n,1} \\ \omega_{n,1} & -\gamma \end{bmatrix}.$$

Therefore, we need to evaluate which value of the natural frequency $\omega_{n,1}$ let us maximize the module of the frequency response $|H_{tot}(j\omega)| = |H_1(j\omega)|$ evaluated at the frequency $\omega = \omega_0$. In particular, in the analysis, ω_0 is fixed.

The optimization problem can be reformulated as follows:

$$\omega_{n,1}^* = \arg\max_{\omega_{n,1}} |H_1(j\omega_0; \omega_{n,1})|. \tag{4.1}$$

Now, given the following frequency response

$$H(j\omega_0; \omega_{n,1}) = K \frac{\gamma + j\omega_0}{(\gamma^2 + \omega_{n,1}^2 - \omega_0^2) + j2\gamma\omega_0}, \tag{4.2}$$

the module is given by

$$|H(j\omega_0; \omega_{n,1})| = K \left(\frac{\omega_0^2 + \gamma^2}{(\gamma^2 + \omega_{n,1}^2 - \omega_0^2)^2 + 4\gamma^2\omega_0^2} \right)^{\frac{1}{2}}. \tag{4.3}$$

and the optimal frequency $\omega_{n,1}^*$ can be found by computing the derivative of the module with respect to $\omega_{n,1}$ and setting it equal to zero:

$$\frac{d}{d\omega_{n,1}} |H(j\omega_0; \omega_{n,1})| = 0. \tag{4.4}$$

For simplicity's sake, we will compute the derivative of the squared module with respect to the natural frequency $\omega_{n,1}$.

$$\frac{d}{d\omega_{n,1}}|H(j\omega_0;\omega_{n,1})|^2 = K^2\frac{-2\omega_{n,1}(\omega_0^2+\gamma^2)(\gamma^2+\omega_{n,1}^2-\omega_0^2)}{\left((\gamma^2+\omega_{n,1}^2-\omega_0^2)^2+4\gamma^2\omega_0^2\right)^2} = 0 \tag{4.5}$$

This equation has a trivial solution $\omega_{n,1} = 0$, which will be overlooked, and a second solution $\omega_{n,1}^*$, that is

$$\omega_{n,1}^* = \sqrt{\omega_0^2-\gamma^2}. \tag{4.6}$$

We can see that the assumption that ω_0 is taken to be much grater than γ, i.e., the assumption that the ratio $\rho' = \gamma/\omega_0 \to 0$, leads to the following approximation

$$\omega_{n,1}^* \approx \omega_0.$$

Remark. Considering Equation (4.6), it seems clear that the optimal value of the natural frequency, i.e., $\omega_{n,1}^*$, is significantly near to the value ω_0, especially when $\rho' = \gamma/\omega_0 \to 0$. This means that, given an input signal with high frequency ω_0, the E-I system Σ_1 must *tune* its natural frequency at the same frequency of the incoming signal, in order to maximize the signal-to-noise ratio, hence the propagation of the signal over the background noise.

Now, in order to understand for which sets of parameters $\{K, \gamma, \omega_{n,1}\}$ we actually obtain an amplification of the signal, we impose that $\rho' \to 0$, hence $\omega_{n,1}^* \approx \omega_0$: using this approximation the expression of the module of the transfer function $|H(j\omega_0;\omega_{n,1}^*)|$ simplifies into the following expression:

$$|H(j\omega_0;\omega_{n,1}^*)| \approx K\left(\frac{\omega_0^2+\gamma^2}{\gamma^4+4\gamma^2\omega_0^2}\right)^{\frac{1}{2}}, \qquad \text{if} \quad \omega_{n,1}^* \approx \omega_0. \tag{4.7}$$

If we take the limit of $|H(j\omega_0;\omega_{n,1}^*)|$ as $\rho = \gamma/\omega_{n,1}^* \approx \gamma/\omega_0 = \rho'$ approaches to zero,

$$\begin{aligned}\lim_{\rho\to 0}|H(j\omega_0;\omega_{n,1}^*)| &= \lim_{\rho\to 0} K\left(\frac{\omega_0^2\left(1+\frac{\gamma^2}{\omega_0^2}\right)}{\omega_0^2\left(\frac{\gamma^2}{\omega_0^2}\gamma^2+4\gamma^2\right)}\right)^{\frac{1}{2}} = \\ &= \lim_{\rho\to 0} K\left(\frac{\omega_0^2\left(1+\rho^2\right)}{\omega_0^2\left(\rho^2\gamma^2+4\gamma^2\right)}\right)^{\frac{1}{2}}\end{aligned} \tag{4.8}$$

we can further simplify the last expression into

$$\lim_{\rho\to 0} |H(j\omega_0;\omega^*_{n,1})| = K\left(\frac{\omega_0^2}{4\gamma^2\omega_0^2}\right)^{\frac{1}{2}} = K\left(\frac{1}{4\gamma^2}\right)^{\frac{1}{2}} = K\frac{1}{2\gamma}. \tag{4.9}$$

In order to have an amplification, $|H(j\omega_0;\omega_{n,1})|$ need to be greater than one: therefore, under the assumption that $\rho \to 0$, we can impose the following inequality

$$|H(j\omega_0;\omega^*_{n,1})| > 1 \iff K\frac{1}{2\gamma} > 1 \iff \gamma < \frac{K}{2}. \tag{4.10}$$

Finally, by recalling that the transfer function gain K was defined as $K = \beta c$, where β and c are the parameters of the input and output matrix, $\boldsymbol{B}_1$ and $\boldsymbol{C}_1$ respectively, we can always impose that the condition above always holds true, for whichever value of γ we choose.

We can also demonstrate that, using the assumption that $\rho \to 0$ is particularly convenient also for the phase of the frequency response $H(j\omega_0;\omega_{n,1})$ evaluated at ω_0, since it is equal to zero.

Indeed, if we impose that $\omega^*_{n,1} = \omega_0$, we can rewrite (4.2) as follows

$$H(j\omega_0;\omega_{n,1}) = K\frac{\gamma + j\omega_0}{\gamma^2 + j2\gamma\omega_0}, \tag{4.11}$$

thus the phase of the frequency response is

$$\begin{aligned}\arg\left(H(j\omega_0;\omega^*_{n,1})\right) &= \arg(K) + \arg(\gamma + j\omega_0) + \arg(\gamma^2 + j2\gamma\omega_0) = \\ &= \arctan\left(\frac{\omega_0}{\gamma}\right) - \arctan\left(\frac{2\gamma\omega_0}{\gamma^2}\right) = \\ &= \arctan\left(\frac{\omega_0}{\gamma}\right) - \arctan\left(\frac{2\omega_0}{\gamma}\right) = \\ &= \arctan\left(\frac{1}{\rho}\right) - \arctan\left(\frac{2}{\rho}\right)\end{aligned}$$

and taking the limit for $\rho \to 0^+$ (since ρ is taken to be strictly positive) finally leads to

$$\begin{aligned}\lim_{\rho\to 0^+}\left[\arg\left(H(j\omega_0;\omega^*_{n,1})\right)\right] &= \lim_{\rho\to 0^+}\left[\arctan\left(\frac{1}{\rho}\right) - \arctan\left(\frac{2}{\rho}\right)\right] = \\ &= \frac{\pi}{2} - \frac{\pi}{2} = 0.\end{aligned}$$

Chapter 5

Interconnection of 2 E-I systems

We will now examine simple cases of interconnections between distinct neural oscillators that might help us to understand the functioning underlying a more general and complex network of oscillators.

We will start by studying the case where two E-I neural oscillators, i.e., Σ_1 and Σ_2, are connected in series and then we will move the focus on the feedback interconnection between two oscillators.

For every configuration we analyze, the aim will be to investigate which set of values of the parameters of each state matrix $\boldsymbol{A}_i$, with $i = 1,2$, maximizes the signal-to-noise ratio, according to the optimization problem we defined and disclosed in Section 3.3. We stress that, in order to simplify the analysis, we will often assume that the parameters γ of each E-I oscillator is set to be much smaller than the frequency of the input signal ω_0 and approximately near to zero.

5.1 Series Connection

Let us start by analyzing a simple case of interconnection between two E-I systems, i.e., Σ_1 with input u_1, output y_1 and natural frequency $\omega_{n,1}$ and Σ_2 with input u_2, output y_2 and natural frequency $\omega_{n,2}$. The series implies that the output of the first system is also the input for the second system, i.e., $u_2 = y_1$. Thus, the series between the two systems Σ_1 and Σ_2 can be

perceived as a system with input u_1 and output y_2 and will be called Σ_s, shown in Figure 5.1. We also assume that the state matrices $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ of the systems Σ_1 and Σ_2, respectively, are characterized by the same value of $\gamma_i = \gamma$, while the natural frequencies $\omega_{n,1}$ and $\omega_{n,2}$ of the systems can vary,

$$\boldsymbol{A}_1 = \begin{bmatrix} -\gamma & -\omega_{n,1} \\ \omega_{n,1} & -\gamma \end{bmatrix}, \qquad \boldsymbol{A}_2 = \begin{bmatrix} -\gamma & -\omega_{n,2} \\ \omega_{n,2} & -\gamma \end{bmatrix}.$$

In this case, the aim will be to evaluate for which values of the natural frequencies $\omega_{n,1}$ and $\omega_{n,2}$ we are able to maximize the module of the frequency response evaluated at $\omega = \omega_0$, $|H_{tot}(j\omega)| = |H_s(j\omega)|$. The optimization problem can be reformulated as follows:

$$\{\omega^*_{n,1}, \omega^*_{n,2}\} = \underset{\omega_{n,1},\omega_{n,2}}{\arg\max} \, |H_s(j\omega_0; \{\omega_{n,1}, \omega_{n,2}\})|. \tag{5.1}$$

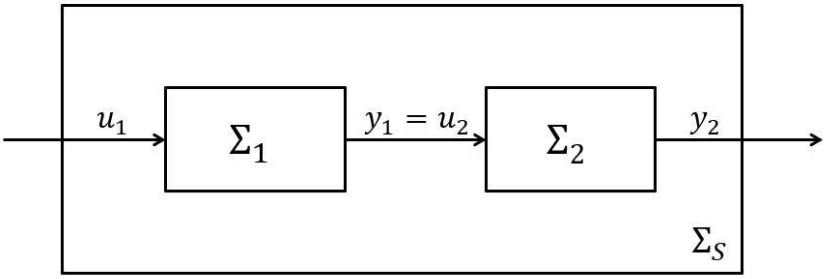


Figure 5.1: A schematic representation of the series between Σ_1 and Σ_2: the output of the first system is also the input of the second system.

The overall transfer function of $H_s(s)$ can be easily computed with the product of the transfer functions $H_1(s)$ and $H_2(s)$ of the single systems

$$H_s(s) = H_1(s) \cdot H_2(s)$$

and the corresponding response function and its module, both evaluated at $\omega = \omega_0$, can be easily derived from the preceding equation:

$$H_s(j\omega_0; \{\omega_{n,1}, \omega_{n,2}\}) = H_1(j\omega_0; \omega_{n,1}) H_2(j\omega_0; \omega_{n,2}) \tag{5.2}$$

and

$$|H_s(j\omega_0; \{\omega_{n,1}, \omega_{n,2}\})| = |H_1(j\omega_0; \omega_{n,1})||H_2(j\omega_0; \omega_{n,2})|. \tag{5.3}$$

In order to solve the optimization problem for the series system Σ_s, it is possible to exploit what was derived in the previous section for the single E-I system, since the two frequency responses are functions only of the respective natural frequency. This means that the optimal natural frequencies $\omega^*_{n,1}$ and $\omega^*_{n,2}$ satisfies

$$\omega^*_{n,1} = \omega^*_{n,2} = \sqrt{\omega_0^2 - \gamma^2},$$

which in turn implies that the two systems must be "synchronized" on the same natural frequency.

In addition, when

$$\rho = \gamma/\omega_0 \to 0$$

the maximum value of $|H_s(j\omega_0; \{\omega_{n,1}, \omega_{n,2}\})|$ is obtained when $\omega^*_{n,1} = \omega^*_{n,2} = \omega_0$. In this case, by simply tuning the two natural frequencies $\omega_{n,1}$ and $\omega_{n,2}$ on the frequency of the input signal ω_0, we obtain the maximum value of the module of the frequency response of the overall system, resulting also in the maximum value of the SNR.

5.2 Feedback Connection

In this section we will now focus on analyzing a different type of interconnection, in particular the feedback interconnection between two E-I systems, i.e. Σ_1, Σ_2. Indeed, the feedback connection appears to be quite explicative and useful when we try to describe the case where an E-I system interfere with another system creating a "loop", as represented below in Figure 5.2.

In order to analyze the more realistic case where the output of the reverse-path system Σ_2 adds up to the input of the forward-path system Σ_1, we will from now on consider only the positive feedback scenario, since we are assuming that the communication among different populations is carried out only by the excitatory connections.

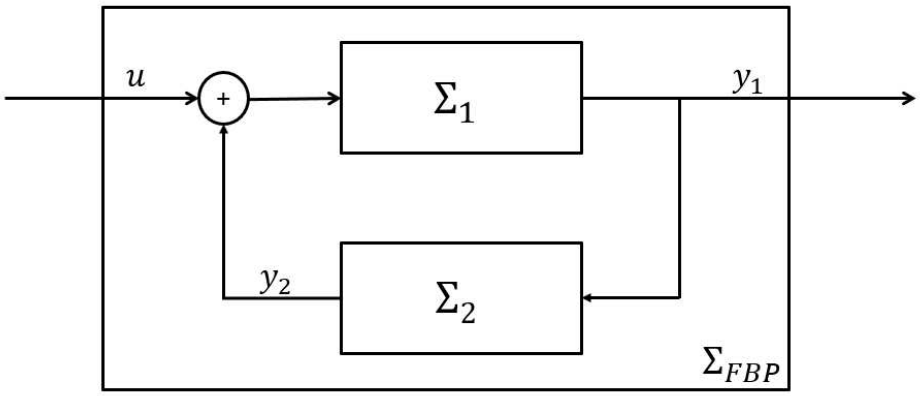


Figure 5.2: Schematic representation of the positive-feedback system that will be analyzed in this section. The feedback signal coming from the system in the reverse path adds up to the incoming signal, thus the name "positive feedback".

The transfer function of the overall system Σ_{FBP} with positive feedback can be easily found as follows:

$$H_{FBP}(s) = \frac{H_1(s)}{1 - H_1(s)H_2(s)}. \tag{5.4}$$

where $H_1(s)$ and $H_2(s)$ are respectively the transfer function of Σ_1 and Σ_2.

The expression above leads us to the following transfer function

$$H_{FBP}(s) = \frac{K_1 \dfrac{(s+\gamma)}{s^2+\omega_{n,1}^2+\gamma^2+2\gamma s}}{1 - K_1K_2\dfrac{(s+\gamma)^2}{(s^2+\omega_{n,1}^2+\gamma^2+2\gamma s)(s^2+\omega_{n,2}^2+\gamma^2+2\gamma s)}}, \quad (5.5)$$

which, after few simplification steps, resolves into

$$\begin{aligned} H_{FBP}(s) &= \frac{n(s)}{d(s)} \\ &= \frac{K_1(s+\gamma)(s^2+\omega_{n,2}^2+\gamma^2+2\gamma s)}{(s^2+\omega_{n,1}^2+\gamma^2+2\gamma s)(s^2+\omega_{n,2}^2+\gamma^2+2\gamma s) - K_1K_2(s+\gamma)^2} \end{aligned} \quad (5.6)$$

First of all, we need to understand for which values of the parameters K_1, K_2, γ, $\omega_{n,1}$ and $\omega_{n,2}$ we actually have a stable system. As we already know, Σ_{FBP} will be stable if the denominator of its transfer function $H_{FBP}(s)$ has roots with negative real part. In order to discuss the stability of Σ_{FBP} we will use the Routh-Hurwitz criterion on the stability of dynamic systems.

5.2.1 Stability of the closed-loop system

The Routh-Hurwitz criterion is a mathematical test that provide a necessary and sufficient condition for the stability of a linear time invariant dynamical system. In particular, the test can be used to determine whether all the roots of the characteristic polynomial of a linear system have negative real parts. If any control system does not satisfy the necessary condition, then we can say that the dynamical system is unstable. But, if the dynamical system satisfies the necessary condition, then it may or may not be stable. So, the sufficient condition is helpful for knowing whether the control system is stable or not.

In this case, the characteristic polynomial corresponds to the denominator of the closed-loop transfer function $H_{FBP}(s)$, that is

$$\begin{aligned} d(s) = s^4 + 5\gamma s^3 &+ (6\gamma^2 + \omega_{n,2}^2 + \omega_{n,1}^2 - K_1K_2)s^2 \\ &+ (4\gamma^3 + 2\gamma(\omega_{n,1}^2 + \omega_{n,2}^2) - 2K_1K_2\gamma)s + \\ &\quad (\gamma^4 + \gamma^2(\omega_{n,1}^2 + \omega_{n,2}^2) + \omega_{n,1}^2\omega_{n,2}^2 - K_1K_2\gamma^2). \end{aligned} \quad (5.7)$$

As we can see, the characteristic polynomial is a fourth-order polynomial in the form

$$d(s) = a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0.$$

The necessary condition states that the coefficients of the characteristic polynomial must be positive, in order to guarantee that all roots are located in the open left half-plane

$$a_i > 0 \quad \text{for} \quad i = 0, \ldots, 4. \tag{5.8}$$

The sufficient condition is that all the elements of the first column of the Routh array should have the same sign, meaning that all the elements of the first column of the Routh array should be either positive or negative. For a fourth-order polynomial, the Routh array correspond to

$$\boldsymbol{R}(d) = \begin{bmatrix} a_4 & a_2 & a_0 \\ a_3 & a_1 & 0 \\ b_3 & b_2 & 0 \\ c_2 & 0 & 0 \\ d_1 & 0 & 0 \end{bmatrix} \tag{5.9}$$

where the elements b_i, c_i and d_i are computed as

$$r_{i,j} = \frac{\begin{vmatrix} r_{i-2,i} & r_{i-2,j+1} \\ r_{i-1,i} & r_{i-1,j+1} \end{vmatrix}}{-r_{i-1,i}}.$$

Thus, the stability of the closed-loop system is guaranteed when

$$a_4, a_3, b_3, c_2, d_1 > 0. \tag{5.10}$$

In particular, the elements on Routh array correspond to:

$$b_3 = \frac{\begin{vmatrix} a_4 & a_2 \\ a_3 & a_1 \end{vmatrix}}{-a_3} = 5\gamma^2 - \frac{K_1 K_2}{2} + \frac{\Omega_{1,2}}{2},$$

$$b_2 = \frac{\begin{vmatrix} a_4 & a_0 \\ a_3 & 0 \end{vmatrix}}{-a_3} = a_0,$$

$$c_2 = \frac{\begin{vmatrix} a_3 & a_1 \\ b_3 & b_2 \end{vmatrix}}{-b_3} = \frac{\gamma}{b_3}\left[(K_1 K_2 - 4\gamma^2)^2 + \Omega_{1,2}^2 + 2\Omega_{1,2}(4\gamma^2 - K_1 K_2)\right],$$

$$d_1 = \frac{\begin{vmatrix} b_3 & b_2 \\ c_2 & 0 \end{vmatrix}}{-c_2} = b_2 = a_0,$$

where we denoted with $\Omega_{1,2}$ the sum of the squared natural frequencies, i.e. $\Omega_{1,2} = \omega_{n,1}^2 + \omega_{n,2}^2$.

Hence, the closed-loop system is stable if

$$\begin{cases} a_4 & = 1 > 0 \\ a_3 & = 4\gamma > 0 \\ b_3 & = 5\gamma^2 - \dfrac{K_1 K_2}{2} + \dfrac{\Omega_{1,2}}{2} > 0 \\ c_2 & = \dfrac{\gamma}{b_3}\left[(K_1 K_2 - 4\gamma^2)^2 + \Omega_{1,2}^2 + 2\Omega_{1,2}(4\gamma^2 - K_1 K_2)\right] > 0 \\ d_1 & = a_0 = \gamma^2 \Omega_{1,2} + \omega_{n,1}^2 \omega_{n,2}^2 + \gamma^2(\gamma^2 - K_1 K_2) > 0 \end{cases} \tag{5.11}$$

Since K_1, K_2, $\gamma > 0$, the conditions above are satisfied $\forall \omega_{n,1}, \omega_{n,2} \geq 0$ if and only if

$$\begin{cases} \gamma > 0 \\ 5\gamma^2 - \dfrac{K_1 K_2}{2} > 0 \\ 4\gamma^2 - K_1 K_2 > 0 \\ \gamma^2 - K_1 K_2 > 0 \end{cases} \iff \quad 0 < \sqrt{K_1 K_2} < \gamma. \tag{5.12}$$

Please notice that this particular condition also satisfies the necessary conditions shown in Equation (5.8). Thus, as we can see, the condition above guarantees that all the poles of its transfer function have negative real part and that the closed-loop system Σ_{FBP} is stable.

5.2.2 Numerical insights on the optimal frequencies

In this case, the optimization problem presented in Section 3.3 will resort to finding for which values of the natural frequencies $\{\omega_{n,1}, \omega_{n,2}\}$ we obtain the maximum value of the module of the frequency response function $H_{FBP}(j\omega)$ when evaluated at $\omega = \omega_0$, i.e.

$$\{\omega^*_{n,1}, \omega^*_{n,2}\} = \arg\max_{\omega_{n,1},\omega_{n,2}}(|H_{FBP}(j\omega_0; \{\omega_{n,1}, \omega_{n,2}\})|) \tag{5.13}$$

where $H_{FBP}(j\omega_0; \{\omega_{n,1}, \omega_{n,2}\})$ can be written as

$$\frac{K_1(\gamma + j\omega_0)(\omega^2_{n,2} - \omega_0^2 + \gamma^2 + j2\gamma\omega_0)}{(\omega^2_{n,1} - \omega_0^2 + \gamma^2 + j2\gamma\omega_0)(\omega^2_{n,2} - \omega_0^2 + \gamma^2 + j2\gamma\omega_0) - K_1K_2(\gamma + j\omega_0)^2}. \tag{5.14}$$

In the closed-loop configuration, the optimization problem appears to be way more difficult to solve than the problem with two E-I systems in series. In order to get some insights on the solution to this problem, we first evaluate where the optimal frequencies are located via numerically methods. Therefore, we prepared a code where, for different values of the natural frequencies of the systems, i.e., $\omega_{n,1}$ e $\omega_{n,2}$, we compute the SNR in response to a sinusoidal signal $u(t) = A_0 \sin(\omega_0 t)$.

We started by creating two arrays of natural frequencies centered around the frequency of the incoming signal, i.e., ω_0. Then, for each combination of the natural frequencies, we computed the SNR through the transfer function of the closed-loop system. The matrix of values of the SNR are then plotted in a colormap with scaled colors, where each hue correspond to a different value of the SNR and each point of the map to a value of the SNR computed with the corresponding natural frequencies on the horizontal and vertical axes. An example of the colormap obtained with the code is shown if Figure 5.3. The code and the values of the parameters chosen for this particular simulation are included in the Appendix D.

Please notice that, in order to get a valid representation of the matrix of SNR values, we need to make sure that the stability condition shown in Equation (5.8) holds true. In this case, for example, we chose $\gamma = 10$, $\omega_0 = 100$, and $K_1 = K_2 = 9$: with these values, the stability condition holds true and the closed-loop system is stable. As we can see from the Figure 5.3, the maximum values of the SNR (which are also the brightest points on the colormap) are condensed in a circular neighbourhood around the optimal

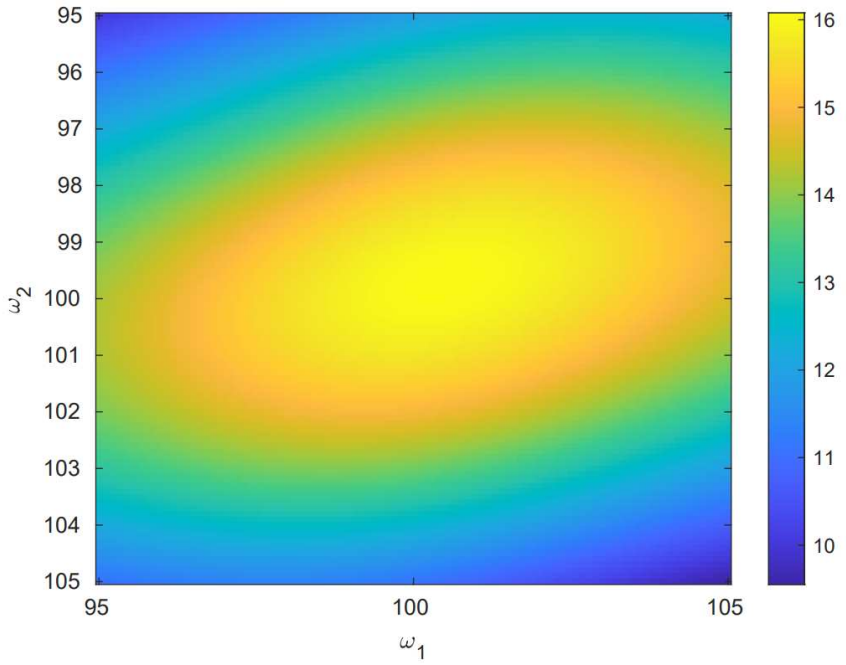


Figure 5.3: Colormap of the values of the SNR found with the simulation.

frequencies $\{\omega^*_{n,1}, \omega^*_{n,2}\}$. In particular, we can also notice that the center of the circular neighbourhood has its coordinates exactly at the center of the values on the horizontal and vertical axes. This also means that we will have the maximum value of the SNR when the natural frequencies are located very closely to each other, i.e., when the two E-I systems are "synchronized". Interestingly, it turns out that the natural frequencies are also close to the frequency of the input signal ($\omega_0 = 100$ in the simulation of Figure 5.3).

Hence, in the next section the main attempt will be focused on investigating the case in which the two natural frequencies are set to be exactly equal to the frequency of the incoming signal, resulting in an overall synchronization of the feedback system. Particular attention will be given to the magnitude of the closed-loop transfer function, which is also directly proportional to the final SNR value.

5.2.3 The case $\omega_{n,1} = \omega_{n,2} = \omega_0$

The case where the two natural frequencies $\omega_{n,1}$ and $\omega_{n,2}$ are taken to be equal seems particularly interesting, since it might happen that both the E-I systems Σ_1 and Σ_2 tune their respective natural frequency on the frequency of the incoming signal, in order to preserve the synchronization, as we discussed previously.

Thus, we can start the analysis by rewriting the transfer function of Σ_{FBP} shown in Equation (5.6) with $\omega_{n,1} = \omega_{n,2} = \omega_n$:

$$H_{FBP}(s) = K_1 \frac{(s+\gamma)(s^2+\omega_n^2+\gamma^2+2\gamma s)}{(s^2+\omega_n^2+\gamma^2+2\gamma s)^2 - K_{12}(s+\gamma)^2} = K_1 \frac{n(s)}{d(s)}, \tag{5.15}$$

where with K_{12} we denoted the product between the two gains K_1 and K_2, with $n(s)$ and $d(s)$ the numerator and denominator of $H_{FBP}(s)$, respectively.

From now, we are going to analyze the numerator and denominator individually, in order to get to the final expressions of the transfer function of the closed-loop system when the two natural frequencies are set to be same.

After some trivial algebraic steps, the numerator $n(s)$ can be simplified into

$$\begin{aligned} n(s) &= (s+\gamma)(s^2+\omega_n^2+\gamma^2+2\gamma s) = \\ &= s^3 + \omega_n^2 s + \gamma^2 s + 2\gamma s^2 + \gamma s^2 + \gamma\omega_n^2 + \gamma^3 + 2\gamma^2 s = \\ &= s^3 + 3\gamma s^2 + (\omega_n^2 + 3\gamma^2)s + \gamma^3 + \omega_n^2\gamma. \end{aligned} \tag{5.16}$$

Then, we can evaluate $n(s)$ at $s = j\omega_0$ in order to obtain the numerator of the frequency response at the particular frequency ω_0

$$n(j\omega_0) = -j\omega_0^3 - 3\gamma\omega_0^2 + j(\omega_n^2 + 3\gamma^2)\omega_0 + \gamma^3 + \omega_n^2\gamma, \tag{5.17}$$

and setting the natural frequencies of the systems equal to ω_0, i.e. $\omega_n = \omega_0$, finally leads us to the following expression

$$\begin{aligned} n(j\omega_0)|_{\omega_n=\omega_0} &= -j\omega_0^3 - 3\gamma\omega_0^2 + j(\omega_0^2 + 3\gamma^2)\omega_0 + \gamma^3 + \omega_0^2\gamma = \\ &= -j\omega_0^3 - 3\gamma\omega_0^2 + j\omega_0^3 + j3\gamma^2\omega_0 + \gamma^3 + \omega_0^2\gamma = \\ &= -2\gamma\omega_0^2 + \gamma^3 + j3\gamma^2\omega_0. \end{aligned} \tag{5.18}$$

On the other hand, the denominator of the transfer function $d(s)$ can be further simplified into

$$\begin{aligned} d(s) = s^4 &+ (4\gamma)s^3 + (6\gamma^2 + 2\omega_n^2 - K_{12})s^2 \\ &+ (4\gamma\omega_n^2 + 4\gamma^3 - 2K_{12}\gamma)s + \omega_n^4 + \gamma^4 + 2\gamma^2\omega_n^2 - K_{12}\gamma^2. \end{aligned} \tag{5.19}$$

In an analogous way, we can evaluate the denominator $d(s)$ at $s = j\omega_0$, in order to obtain the denominator of the frequency response evaluated at the frequency of interest ω_0

$$d(j\omega_0) = \omega_0^4 - j(4\gamma)\omega_0^3 - (6\gamma^2 + 2\omega_n^2 - K_{12}^2)\omega_0^2 + j(4\gamma\omega_n^2 + 4\gamma^3 - 2K_{12}^2\gamma)\omega_0 + \omega_n^4 + \gamma^4 + 2\gamma^2\omega_n^2 - K_{12}^2\gamma^2, \quad (5.20)$$

and finally, by setting the natural frequencies of the E-I systems equal to ω_0, i.e. $\omega_n = \omega_0$, we obtain the following expression:

$$d(j\omega_0)|_{\omega_n=\omega_0} = \omega_0^4 - j(4\gamma)\omega_0^3 - (6\gamma^2 + 2\omega_0^2 - K_{12})\omega_0^2 + j(4\gamma\omega_0^2 + 4\gamma^3 - 2K_{12}\gamma)\omega_0 + \omega_0^4 + \gamma^4 + 2\gamma^2\omega_0^2 - K_{12}\gamma^2, \quad (5.21)$$

which results into

$$d(j\omega_0)|_{\omega_n=\omega_0} = (4\gamma^2 - K_{12}) - K_{12}\gamma^2 + j(2K_{12}\gamma - 4\gamma^3)\omega_0. \quad (5.22)$$

Thus, combining the expressions in equations (5.18) and (5.22), the final expression of the frequency response function $H_{FBP}(j\omega)$ of the positive-feedback system Σ_{FBP} evaluated at the frequency of the input signal ω_0 and with $\omega_n = \omega_0$ results into

$$H_{FBP}(j\omega_0)|_{\omega_n=\omega_0} = K_1 \frac{2\gamma\omega_0^2 - \gamma^3 + j3\gamma^2\omega_0}{(4\gamma^2 - K_{12})\omega_0^2 + K_{12}\gamma^2 + j(2K_{12}\gamma - 4\gamma^3)\omega_0}. \quad (5.23)$$

If γ is taken to be much less than the input frequency ω_0, i.e.,

$$\rho = \gamma/\omega_0 \to 0$$

the expression shown in Equation (5.23) can be further simplified both at the numerator and denominator by ignoring the terms that are not function of ω_0, leading us to

$$H_{FBP,simp}(j\omega_0)|_{\omega_n=\omega_0} = K_1 \frac{2\gamma\omega_0^2 + j3\gamma^2\omega_0}{(4\gamma^2 - K_{12})\omega_0^2 + j(2K_{12}\gamma - 4\gamma^3)\omega_0}. \quad (5.24)$$

Lastly, by dividing for ω_0 both at the numerator and denominator, we reach the final expression of the frequency response function we were seeking

$$H_{FBP,simp}(j\omega_0)|_{\omega_n=\omega_0} = K_1 \frac{2\gamma\omega_0 + j3\gamma^2}{(4\gamma^2 - K_{12})\omega_0 + j(2K_{12}\gamma - 4\gamma^3)}. \quad (5.25)$$

We are now able to compute the module of the frequency response function shown in Equation (5.25), in order to find for which set of values of the parameters γ, K_1, K_2 we actually have an amplification of the incoming signal.

The module of the frequency response function $H_{FBP,simp}(j\omega_0)|_{\omega_n=\omega_0}$ shown in Equation (5.25) can be easily computed and results in

$$|H_{FBP,simp}(j\omega_0)|_{\omega_n=\omega_0}| = K_1 \left(\frac{4\gamma^2\omega_0^2 + 9\gamma^4}{(4\gamma^2 - K_{12})^2\omega_0^2 + (2K_{12}\gamma - 4\gamma^3)^2} \right)^{\frac{1}{2}} \tag{5.26}$$

which, under the initial hypothesis that $\gamma \ll \omega_0$, can be further simplified into

$$|H_{FBP,simp}(j\omega_0)|_{\omega_n=\omega_0}| = K_1 \left(\frac{4\gamma^2\omega_0^2}{(4\gamma^2 - K_{12})^2\omega_0^2} \right)^{\frac{1}{2}} \tag{5.27}$$

and finally into

$$|H_{FBP,simp}(j\omega_0)|_{\omega_n=\omega_0}| = K_1 \left(\frac{4\gamma^2}{(4\gamma^2 - K_{12})^2} \right)^{\frac{1}{2}}. \tag{5.28}$$

According to the signs of K_{12} and γ we can split the equation above into the following system:

$$|H_{FBP,simp}(j\omega_0)|_{\omega_n=\omega_0}| = \begin{cases} K_1 \dfrac{2\gamma}{(4\gamma^2 - K_{12})} & \text{if} \quad 0 < \sqrt{K_{12}} < 2\gamma \\ \infty & \text{if} \quad \sqrt{K_{12}} = 2\gamma \\ -K_1 \dfrac{2\gamma}{(4\gamma^2 - K_{12})} & \text{if} \quad \sqrt{K_{12}} > 2\gamma. \end{cases} \tag{5.29}$$

As we can see, when $\sqrt{K_{12}}$ approaches the value 2γ, we are more likely to have an amplification of the incoming signal. However, we also need to recall the stability condition of Equation (5.8) we found in the previous section: in fact,

$$0 < \sqrt{K_{12}} < \gamma$$

is a even stricter condition and this means that, under the stability condition, the module of the frequency response function turns out to be

$$|H_{FBP,simp}(j\omega_0)|_{\omega_n=\omega_0}| = K_1 \frac{2\gamma}{(4\gamma^2 - K_{12})} \iff 0 < \sqrt{K_{12}} \leq \gamma. \tag{5.30}$$

As we can derive from the equation above, the most reasonable and optimal choice of the parameters K_1 and K_2 is to set their product K_{12} equal to γ^2, in order to have the maximum amplification possible while stabilizing the closed-loop system.

On the other hand, the stability condition may vary and be slightly wider if we assume that $\Omega_{1,2} = \omega_{n,1}^2 + \omega_{n,2}^2 \gg \gamma$, K_{12}. In this case, the equations shown in (5.11) hold true $\forall \omega_{n,1}, \omega_{n,2}$ only if $4\gamma^2 - K_1 K_2$, and thus $0 < \sqrt{K_{12}} < 2\gamma$. This might also be explanatory of the results presented in (5.29), since we assumed that $\omega_n = \omega_n \gg \gamma$. Consequently, under the main assumption that $\Omega_{1,2} = \omega_{n,1}^2 + \omega_{n,2}^2 \gg \gamma$, K_{12}, the optimal choice for the systems gains turns out to be slightly different: in fact, we will obtain the maximum amplification possible when K_{12} is set to be equal to $4\gamma^2$.

Chapter 6

General Interconnection of N E-I Systems

In this chapter we will examine the more generic case where we have N interconnected E-I systems. Similarly to what we outlined in the previous chapter, after deriving the general equations that describe the dynamics of the network, we investigate its stability by taking into account its adjacency matrix and the relationship between the latter and the overall state matrix of network. Then, in order to extend the implications and the results that were found and highlighted in the previous sections, we will attempt to solve the optimization problem for particular configurations, i.e., the series of N oscillators and networks of oscillators built upon specific types of graphs, i.e. directed acyclic graph.

6.1 Equations of the Network

In this chapter we will consider the more generic case where N systems are interconnected to form a network of oscillator systems Σ_i with $i = 1, \ldots, N$ and $N > 2$, as those described in equations (6.1):

$$\Sigma_i : \begin{cases} \dot{x}_i = \boldsymbol{A}_i x_i + \boldsymbol{B}_i u_i \\ \\ y_i = \boldsymbol{C}_i x_i. \end{cases} \tag{6.1}$$

where $\boldsymbol{A}_i = \begin{bmatrix} -\gamma_i & -\omega_i \\ \omega_i & -\gamma_i \end{bmatrix} \in \mathbb{R}^{2\times 2}$, $\boldsymbol{B}_i = \begin{bmatrix} \beta_i \\ 0 \end{bmatrix} \in \mathbb{R}^{2\times 1}$ and $\boldsymbol{C}_i = \begin{bmatrix} c_i & 0 \end{bmatrix} \in \mathbb{R}^{1\times 2}$ are the state matrix, the input matrix and the output matrix of the i-th system Σ_i, respectively.

We will start by assuming that the systems $\Sigma_1, \dots, \Sigma_N$ define the nodes of a directed graph $\Gamma(\mathcal{S}, \mathcal{E})$, where $\mathcal{S} = \{1, \dots, N\}$ is the set of nodes of the graph, with node i corresponding to system Σ_i, and $\mathcal{E}$ the set of edges. The graph Γ can be characterized by its adjacency matrix $\mathcal{H} \in \mathbb{R}^{N\times N}$, which is going to be binary, according to the edges $\mathcal{E}$, and in general asymmetric, because we are considering directed graphs. For more details, please see Appendix A.

The aim will be to reach a compact expression for the state matrix, input and output matrices of the network of systems described over the graph Γ.

In particular, the equations we are seeking will be in the following form:

$$\Sigma^{\Gamma} : \begin{cases} \dot{x}_{tot} = \boldsymbol{A}_{tot} x_{tot} + \boldsymbol{B}_{tot} u_{tot} \\ \\ y_{tot} = \boldsymbol{C}_{tot} x_{tot}, \end{cases} \tag{6.2}$$

where we denoted with $x_{tot} \in \mathbb{R}^{2N}$ the vector that contains all the state vectors of the nodes of the network, i.e. $x_{tot} = \{x_1, \dots, x_{k-1}, x_k, \dots, x_N\}$, with $u_{tot} \in \mathbb{R}$ and $y_{tot} \in \mathbb{R}$ the input and output of the network, respectively. We also recall that each state vector will be composed of two components, i.e. $x_k = \{x_{k,e}, x_{k,i}\}^{\intercal}$ representing the temporal evolution of the excitatory and inhibitory population of the system Σ_k, respectively.

In this case, $\boldsymbol{A}_{tot} \in \mathbb{R}^{2N\times 2N}$ will be the new state matrix of the network, $\boldsymbol{B}_{tot} \in \mathbb{R}^{2N\times 1}$ the new input matrix and $\boldsymbol{C}_{tot} \in \mathbb{R}^{1\times 2N}$ the new output matrix.

First, we are going to consider the k-th node of the network, corresponding to system Σ_k: we suppose that it has in-degree $d_{in}(k) = h$, meaning that it has h inputs coming from the nodes at the previous step $k-1$, i.e. $y^1_{k-1}, \dots, y^h_{k-1}$ and out-degree $d_{out}(k) = 1$, i.e. one output, i.e., y_k, as depicted in Figure 6.1.

In particular, each input y_{k-1} can be rewritten as the product of the state vector x_{k-1} with the output matrix $\boldsymbol{C}_{k-1}$ of the respective system Σ_{k-1}. Each input y_{k-1} will be then multiplied by the input matrix $\boldsymbol{B}_k$ of the k-th system

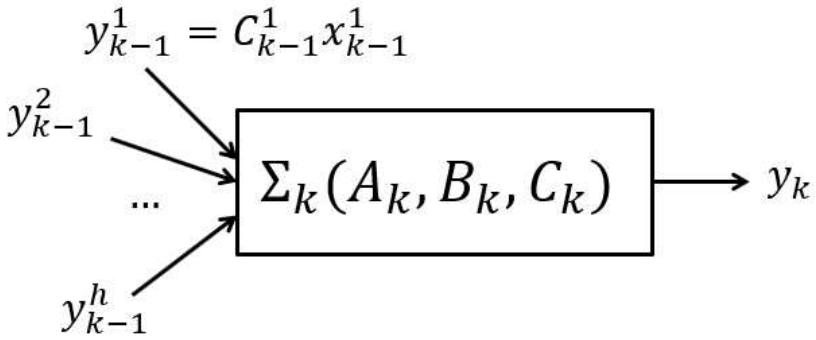


Figure 6.1: Schematic representation of the k-th node Σ_k.

Σ_k. Thus, the overall input u_k can be formulated as

$$u_k = \boldsymbol{B}_k y^1_{k-1} + \cdots + \boldsymbol{B}_k y^h_{k-1} = \boldsymbol{B}_k \sum_{j=1}^{h} \boldsymbol{C}^j_{k-1} x^j_{k-1}, \tag{6.3}$$

while the equations describing the system Σ_k will result into

$$\Sigma_k : \begin{cases} \dot{x}_k = \boldsymbol{A}_k x_k + \boldsymbol{B}_k \sum_{j=1}^{h} \boldsymbol{C}^j_{k-1} x^j_{k-1} \\ \\ y_k = \boldsymbol{C}_k x_k. \end{cases} \tag{6.4}$$

Assuming that the input and output matrices are the same for all the nodes, i.e. $\boldsymbol{B}_k = \boldsymbol{B}$ and $\boldsymbol{C}_k = \boldsymbol{C}$ for $k = 1, \ldots, N$, we can further simplify the equations above into

$$\Sigma_k : \begin{cases} \dot{x}_k = \boldsymbol{A}_k x_k + \boldsymbol{B}\boldsymbol{C} \sum_{j=1}^{h} x^j_{k-1} \\ \\ y_k = \boldsymbol{C}_k x_k. \end{cases} \tag{6.5}$$

Now it is possible to exploit the adjacency matrix $\mathscr{H}$, since it displays in an analytical way the topology and the connections between the several nodes of the graph Γ. In fact, whenever we have a connection between two nodes, for example Σ_k and Σ_{k-1}, the element $\hbar_{k-1,k}$ will be equal to one, if we have an edge that goes from node $k-1$ to node k.

In order to define the matrices of the overall network we are going to introduce a mathematical tool, the so-called *Kronecker product*.

Definition 6.1.1 (Kronecker product)**.** Given a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ and a matrix $\boldsymbol{B} \in \mathbb{R}^{p\times q}$, the Kronecker product between $\boldsymbol{A}$ and $\boldsymbol{B}$ is the $np \times mq$ matrix $\boldsymbol{A} \otimes \boldsymbol{B}$ given by

$$\begin{bmatrix} a_{11}\boldsymbol{B} & \dots & a_{1n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{n1}\boldsymbol{B} & \dots & a_{nm}\boldsymbol{B} \end{bmatrix}.$$

This particular product between matrices will let us compact the expressions above. In fact the state matrix of the network $\boldsymbol{A}_{tot}$ can be formulated as

$$\boldsymbol{A}_{tot} = \operatorname{diag}(\boldsymbol{A}_1, \dots, \boldsymbol{A}_N) + \mathscr{H} \otimes (\boldsymbol{BC}), \tag{6.6}$$

where we can see that the state matrices of the single systems occupy the main diagonal, while the term $\boldsymbol{BC}$ is distributed along the overall state matrix according to the adjacency matrix.

As to the input and output matrices $\boldsymbol{B}_{tot}$ and $\boldsymbol{C}_{tot}$, they will be given by

$$\boldsymbol{B}_{tot} = e_{in} \otimes \boldsymbol{B}, \quad \boldsymbol{C}_{tot} = e_{out}^{\top} \otimes \boldsymbol{C} \tag{6.7}$$

where we denoted with *in* and *out* the input and output node, respectively, and with e the standard basis vector in $\mathbb{R}^N$.

Remark. Please notice that the choice of the input and output nodes is still arbitrary and is going to be a degree of freedom when simulating what happens at the network for different sets of the adjacency matrix, that is also the topology of the network. The most interesting case will be the one where the input and output nodes are chosen to be the furthest possible.

Now that we have obtained the matrices of the state-space representation of the network, we can compute the transfer function of the network, that is

$$H_{tot}^{\Gamma}(s) = \boldsymbol{C}_{tot} \left(s\boldsymbol{I} - \boldsymbol{A}_{tot}\right)^{-1} \boldsymbol{B}_{tot} \tag{6.8}$$

where we denoted with $\boldsymbol{I}$ the identity matrix of size $2N$.

In the next section we will analyze what happens to network when we have different topologies, that is different adjacency matrices, also depending on the choice of the input and output nodes and on the choice of the natural frequencies of the nodes. To do so, as we will see, we will need to run some simulations.

6.2 Stability of the Network

In order to study the behaviour of the network, we have to understand beforehand when and under which particular conditions the network is actually "stable". In fact, we are only interested in the scenario in which our network of oscillators is actually capable of amplifying in a stable way the input signal, without eventually diverging to infinity.

To study the stability of the network we will now need to focus on the state matrix of the network, i.e., $\boldsymbol{A}_{tot}$. As we have seen in the previous section, the state matrix of the network $\boldsymbol{A}_{tot}$ is defined as

$$\begin{bmatrix} \boldsymbol{A}_{11} & \dots & \boldsymbol{A}_{1N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{A}_{N1} & \dots & \boldsymbol{A}_{NN} \end{bmatrix}.$$

where the matrix block on the main diagonal are exactly the state matrices of the nodes of the network, while the elements that are outside the diagonal derive from the product between the adjacency matrix $\mathcal{H}$ and the matrix $\boldsymbol{BC}$.

As a consequence, the terms that do not belong to the main diagonal can be written as

$$\boldsymbol{A}_{ij} = \begin{cases} \boldsymbol{O}_{2\times 2} & \text{if} \quad \{i,j\} \notin \mathcal{E}, i \neq j \\ \boldsymbol{BC} & \text{if} \quad \{i,j\} \in \mathcal{E}, i \neq j \end{cases}$$

More specifically, in order to study the stability of the network, we will be interested in analyzing the *spectrum* of the state matrix, i.e. the set of its eigenvalues,

$$\lambda(\boldsymbol{A}_{tot}) = \{\lambda_1, \lambda_2, \dots, \lambda_{2N},\}$$

since the stability of the network is strictly correlated to them, more particularly to the sign of their real part.

However, the real problem arises when we try to analytically compute the eigenvalues of such a matrix. In fact, $\boldsymbol{A}_{tot}$ is defined as a square block matrix in $\mathbb{R}^{2N}$, meaning that even with two nodes the computation of the eigenvalues of the matrix can get trickier, and eventually almost impossible to solve by hand for greater values of N.

Nevertheless, we can still try to set our attention to some particular cases, in which the computational problem can be easily overcome or bypassed. We will omit the trivial case where all the block matrices outside the main

diagonal are equal to $\boldsymbol{O}_{2\times 2}$, since it would imply that we have no connections between the nodes.

One of the simplest and yet interesting study case is when the state matrix $\boldsymbol{A}_{tot}$ is a triangular block matrix. More specifically, the block matrix is said to be upper (lower) triangular when the block matrices under (above) the main diagonal are all equal to the zero matrix.

For example,

$$\boldsymbol{A}_{tot} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \dots & \boldsymbol{A}_{1N} \\ \boldsymbol{O}_{12} & \boldsymbol{A}_{22} & \dots & \boldsymbol{A}_{2N} \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{O}_{N1} & \boldsymbol{O}_{N2} & \dots & \boldsymbol{A}_{NN} \end{bmatrix}$$

is an upper triangular block matrix $\boldsymbol{A}_{tot}$.

Remark. These types of state matrices can be obtained from a particular form of the adjacency matrix, which is in turn related to a particular class of digraphs, more specifically acyclic digraphs. These last class of digraphs will be thoroughly described and studied in the a subsequent section, where we will focus on interconnection patterns of the network based over acyclic digraphs.

Furthermore, it is possible to prove that the eigenvalues of a triangular block matrix are exactly the eigenvalues of the matrices on the main diagonal. In fact, if we suppose that A_{tot} is a triangular block matrix, its determinant can be written as

$$\det(\boldsymbol{A}_{tot}) = \prod_{i=1}^{N} \det(\boldsymbol{A}_i).$$

As a consequence, also the matrix $\boldsymbol{A}_{tot} - \lambda \boldsymbol{I}$ is a triangular block matrix and thus, by solving the following equation

$$\det(\boldsymbol{A}_{tot} - \lambda \boldsymbol{I}) = \prod_{i=1}^{N} \det(\boldsymbol{A}_i - \lambda \boldsymbol{I}) = 0,$$

we can finally obtain the eigenvalues of $\boldsymbol{A}_{tot}$.

It appears also clear that, since

$$\lambda(\boldsymbol{A}_{tot}) = \bigcup_{i=1}^{N} \lambda(\boldsymbol{A}_i),$$

we will simply need to compute the eigenvalues of the N matrices on the main diagonal, i.e., the state matrices of each node, in order to obtain the eigenvalues of $\boldsymbol{A}_{tot}$, i.e., the state matrix of the overall network.

Remark. Please notice that this simplification is feasible only because the starting matrix is a triangular block matrix. Although this might seem quite a drastic simplification of the network, it lets us exploit all the results we have presented and discussed in the previous chapters.

In fact, we already know that, according to Equation (2.26), the couple of eigenvalues of each state matrix $\boldsymbol{A}_i$ are

$$\lambda_{1,2}^i = -\gamma_i \pm \omega_{n,i},$$

where we recall that γ_i is assumed to be the *damping* parameter of the i-th neural population and $\omega_{n,i}$ its natural frequency.

In the definition of our model, the parameters γ_i are always taken to be strictly positive, since negative values of this parameter would result into an unrealistic scenario. As a direct consequence, the real part of each eigenvalue we have just defined is always strictly negative, i.e.

$$\mathrm{Re}\{\lambda_{1,2}^i\} = -\gamma_i < 0,$$

resulting in an overall stability of the single i-th node. Since the eigenvalues of the network are defined as the set that includes all the eigenvalues of the single nodes composing the network, we will clearly obtain a stable network whenever its state matrix is defined as a triangular block matrix.

For more general interconnection patterns, the stability of the overall system can be assessed via numerical evaluation of the eigenvalues of $\boldsymbol{A}_{tot}$.

6.3 Series of N E-I Systems

Now, let us suppose that we have N E-I systems connected in series, as shown in Figure 6.2. In order to solve the optimization problem formulated in Section 3.3, we can extend what we found in Section 5.1 for the simpler case with two interconnected systems.

The overall transfer function of the system Σ_S can be found by simply multiplying the transfer functions of the single E-I systems,

$$H_{tot}(s) = H_s(s) = H_1(s)H_2(s)\dots H_N(s) = \prod_{i=1}^{N} H_i(s), \tag{6.9}$$

where we recall that each transfer function $H_i(s)$ is also function of the natural frequency $\omega_{n,i}$ of the corresponding E-I system. In this case, we will need to find for which values of the natural frequencies $\{\omega_{n,i}\}_{i=1,\dots,N}$ we obtain the maximum amplification of the incoming sinusoidal signal, hence the maximum value of $\|H_{tot}(j\omega_0)\|$, i.e.

$$\{\omega^*_{n,1},\dots,\omega^*_{n,N}\} = \arg\max_{\{\omega_{n,i}\}}(|H_{tot}(j\omega_0;\{\omega_{n,i}\})|),$$

where $\omega^*_{n,i}$ are the optimal frequencies.

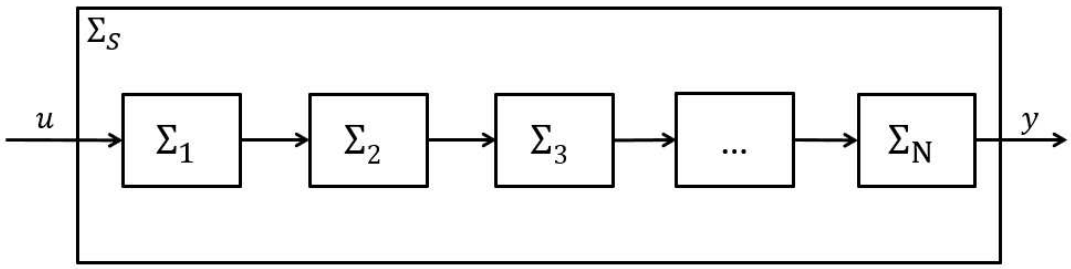


Figure 6.2: Schematic representation of the system Σ_S with N E-I systems interconnected in series.

Assuming also that the incoming signal is a sinusoidal signal $u(t) = A_0 \sin\omega_0 t$, the output signal $y(t)$ can be easily expressed as

$$y(t) = A_0\,|H_{tot}(j\omega_0;\{\omega_{n,i}\})|\sin\left[\omega_0 t + \arg(H_{tot}\,(j\omega_0;\{\omega_{n,i}\}))\right], \tag{6.10}$$

where

$$|H_{tot}(j\omega_0; \{\omega_{n,i}\})| = \left| \prod_{i=1}^{N} H_i(j\omega_0; \{\omega_{n,i}\}) \right|$$

and

$$\arg(H_{tot}(j\omega_0; \{\omega_{n,i}\})) = \sum_{i=1}^{N} \arg(H_i(j\omega_0; \{\omega_{n,i}\})).$$

Similarly to what done before, we can exploit the analysis of the single E-I system to find solve the above optimization problem, since $|H_{tot}(j\omega_0; \{\omega_{n,i}\})|$ is the product of the modules of the frequency responses of the N individual E-I systems. In particular in the limit case $\rho = \gamma/\omega_0 \to 0$ we can easily prove that, in order to maximize the module of the overall frequency response function, we will only need to *tune* each natural frequency $\omega_{n,1}, \ldots, \omega_{n,N}$ on the frequency of the incoming signal ω_0.

In fact, since

$$\left| \prod_{i=1}^{N} H_i(j\omega_0; \omega_{n,i}) \right| = \prod_{i=1}^{N} |H_i(j\omega_0; \omega_{n,i})|$$

we can reformulate the optimization problem as

$$\{\omega^*_{n,1}, \ldots, \omega^*_{n,N}\} = \arg\max_{\{\omega_{n,i}\}} (\prod_{i=1}^{N} |H_i(j\omega_0; \omega_{n,i})|).$$

It appears clear that the product of the frequency response functions will have its maximum value when each frequency response function is tuned through their respective natural frequency exactly on the resonance peak value. In other words, this happens when all the nodes have its natural frequency equal to the frequency of the incoming signal resulting in an overall synchronization of the E-I systems. In mathematical terms, in this case, the optimization problem for the series of N E-I systems is solved for

$$\omega_{n,1} = \omega_{n,2} = \cdots = \omega_{n,N} = \omega_0.$$

In fact, by doing so, each E-I system will be able to provide the maximum amplification possible. Furthermore, if we assume that each E-I oscillator of the series have the same parametrization and hence can be described through the same transfer function $\tilde{H}(s)$, we can show that the amplification has an exponential growth with N.

In fact, let N be the number of equal E-I oscillators with transfer function $\tilde{H}(s)$, then the output signal $y(t)$ will result into

$$y(t) = A_0|\tilde{H}(j\omega_0)|^N \sin\Big[\omega_0 t + N \arg(\tilde{H}(j\omega_0))\Big]. \tag{6.11}$$

Remark. Please note that, with the optimal choice of the natural frequencies

$$\{\omega^*_{n,1}, \ldots, \omega^*_{n,N}\},$$

we will also be able to prevent the series system from introducing an excessive delay into the output signal $y(t)$: in fact, with this particular choice, each E-I system will be able to maintain its phase approximately near to zero, as we derived in Section 5.1. As a result, the phase of the overall transfer function will be constrained and the synchronization in frequency might as well give rise to an overall synchronization in time.

6.4 Interconnection Pattern over Acyclic Graph

In this section we will present and analyze the particular case where the interconnections between the E-I oscillators of the network is based upon a directed acyclic graph. In particular, an *acyclic graph* is a graph without cycles or loops: in other words, when following the graph edges from node to node, you will never run into the same node twice. More specifically, a *directed acyclic graph* (or acyclic digraph or "DAG") is an acyclic graph in which the edges have a precise orientation, as well as a lack of cycles.

An example of a directed acyclic graph with five nodes is shown in Figure 6.3, whose adjacency matrix $\mathcal{H}$ corresponds to

$$\mathcal{H} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

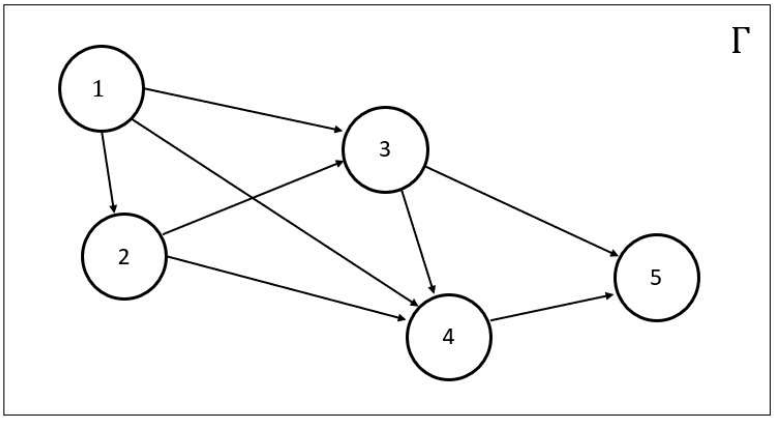


Figure 6.3: Example of a digraph acyclic graph with five nodes.

In particular, the following lemma provides an interesting condition on the adjacency matrix in order to understand if the graph is acyclic or not.

Lemma 6.4.1. *Let Γ be a directed graph with adjacency matrix $\mathcal{H}$ and let N be the number of nodes of Γ. If $\mathcal{H}$ is nilpotent, i.e. $\mathcal{H}^k = 0$ for $k \leq N$, then the directed graph Γ is acyclic.*

We can provide an intuitive proof to it, by recalling that the powers of the adjacency matrix have quite an interesting interpretation: in fact, the

element $\hbar_{ij}^k$ of the matrix $\mathcal{H}^k$ gives us the number of walks of length k from node i to node j. If we assume that $\mathcal{H}$ is nilpotent, then there exists an integer number $k \leq N$, also called *nilpotency index*, so that $\mathcal{H}^k = 0$: this also means that there are no walks of length k or even longer in the graph Γ.

Conversely, if $\mathcal{H}^N \neq 0$, there certainly is a walk of length N and since there are N nodes in the graph, then the same node must appear on the walk more than once, thus creating a cycle.

In particular, any n-dimensional strictly triangular matrix is also nilpotent. Therefore, whenever we have an adjacency matrix which is strictly triangular, the graph derived from the adjacency matrix is going to be acyclic. The contrary in not necessarily true: in fact, we might have a nilpotent adjacency matrix which is not strictly triangular. However, it is possible to demonstrate that the adjacency matrix of an acyclic graph can always be transformed into a strictly triangular matrix through a permutation matrix $\boldsymbol{P}$ (which simply corresponds to a relabelling of the nodes of the network), so that $\boldsymbol{P}^\intercal \mathcal{H} \boldsymbol{P} = \mathcal{H}'$, where $\mathcal{H}'$ is exactly a strictly triangular matrix.

As a result, it is possible to prove that, given an acyclic graph Γ with N nodes and with adjacency matrix $\mathcal{H}$, the eigenvalues of the overall state matrix $\boldsymbol{A}_{tot}$ are exactly the eigenvalues of the state matrices $\boldsymbol{A}_i$.

From now on, we will only take into account adjacency matrices that are stricly triangular, in order to present an analytical solution to the optimization problem. However it is possible to extend the analysis to any kind of nilpotent adjacency matrix by computing a suitable permutation matrix, as explained above.

For example, the state matrix $\boldsymbol{A}_{tot}$ of the network based on the graph shown in Figure 6.3 can be computed through Equation (6.6), leading us to

$$\boldsymbol{A}_{tot} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{BC} & \boldsymbol{BC} & \boldsymbol{BC} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{A}_2 & \boldsymbol{BC} & \boldsymbol{BC} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{A}_3 & \boldsymbol{BC} & \boldsymbol{BC} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{A}_4 & \boldsymbol{BC} \\ \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{A}_5 \end{bmatrix}$$

where $\boldsymbol{A}_i = \begin{bmatrix} -\gamma & -\omega_i \\ \omega_i & -\gamma \end{bmatrix}$ is the state matrix of the i-th node, $\boldsymbol{BC}$ is exactly the matrix given by the product between the vectors $\boldsymbol{B}$ and $\boldsymbol{C}$ and $\boldsymbol{O}$ is the zero matrix $\boldsymbol{O}_{2\times 2}$.

Remark. Please note that, for consistency's sake with the previous chapters, we will assume that, for each node of the network, $\boldsymbol{B} = \begin{bmatrix} \beta & 0 \end{bmatrix}^\intercal$ and $\boldsymbol{C} = \begin{bmatrix} c & 0 \end{bmatrix}$: this choice will let us take into consideration only the case in which the excitatory subgroups of each neural population is responsible for transmitting signals and communicating with other neural populations.

Under this assumption, the product $\boldsymbol{BC}$ results into

$$\boldsymbol{BC} = \begin{bmatrix} \beta \\ 0 \end{bmatrix} \begin{bmatrix} c & 0 \end{bmatrix} = \begin{bmatrix} \beta c & 0 \\ 0 & 0 \end{bmatrix}.$$

We can clearly see that, since the state matrix of the network $\boldsymbol{A}_{tot}$ is a block triangular matrix, its eigenvalues are exactly the eigenvalues of the single state matrices on the main diagonal. As a result, we can infer that any network of E-I oscillators built upon a directed acyclic graph is stable whenever each E-I oscillator of the network is stable itself, which by definition is true, since γ is always taken to be strictly positive.

Furthermore, it is also possible to show that each walk of a directed acyclic graph can be thought of as a *series* of different nodes of precise length. In fact, by definition, a directed acyclic graph does not comprise any loop or cycle and every edge has a precise orientation, making it possible to define a topological order of the nodes. And since any directed acyclic graph can be reduced to a set of directed walks, which per se are simply sequences of nodes in which each node appear only once, we might be able to exploit the results we presented for the series interconnection, in order to solve the optimization problem and find the optimal frequencies in the DAG configuration.

Now, let Γ be a directed acyclic graph with N nodes and let $\mathcal{N}$ be the network of N E-I oscillators built upon Γ. Let us now choose two different nodes u and v of Γ, such that u is the input node, i.e. the node that receives the input signal $u(t)$, and v is the output node, i.e. the node that emits the corrupted output signal $\tilde{y}(t)$. We also suppose that v is *reachable* by u, meaning that there certainly exists at least one directed walk starting from u and ending in v. If not, the output signal is automatically set to zero and any choice the optimization problem is not interesting since $H_{tot}(s) = 0$.

Under these assumptions, the optimization problem results into finding which values of the natural frequencies maximize the module of the frequency response function $H_{tot}(j\omega_0)$ of the network evaluated at ω_0:

$$\{\omega^*_{n,1}, \dots, \omega^*_{n,N}\} = \underset{\{\omega_{n,i}\}_{i=1,\dots,N}}{\arg\max} \left(H_{tot}(j\omega_0)\right)$$

where we recall that $H_{tot}(j\omega_0)$ is obtained from the network transfer function $H_{tot}(s)$ computed through Equation (6.8). And since any directed walk from u to v can be though as a determined set of nodes starting from u and ending in v where all the nodes from u to v are interconnected in series, we can deduce that the optimization problem is solved when all the E-I oscillators are synchronized and tuned on the frequency of the incoming signal $u(t)$. In other words, the optimal natural frequencies that maximize the SNR are

$$\omega_{n,1} = \omega_{n,2} = \cdots = \omega_{n,N} = \omega_0.$$

In this case of degree of amplification depends on the length of the walks connecting u to v. More precisely, the longer such walks are, the greater the amplification is. With this in mind, Equation (6.11) appears to be quite exemplary: in fact, the module of the output signal $|y(t)|$ shows an exponential growth with N, meaning that increasing the number of nodes in the series interconnection gives rise to a drastic amplification of the input signal. In the case of a DAG, the amplification is dictated by the walk (or the walks) with maximum length k_{max}.

We also provide a code to validate our findings, i.e., that the frequency synchronization of the nodes solve the optimization problem we outlined. The code can be found in Appendix E with some comments on the parameters and the variables involved. In particular, the algorithm is designed to find the optimal frequencies of a network of $N = 6$ E-I oscillators built upon any type of topology, that is given by the adjacency matrix. The code can be easily extended to larger networks comprising more than six nodes: however, finding the optimal values of the natural frequencies needs some heavy computation and increasing the number of nodes of the network is likely to drastically increase the running time of the code. We will now summarize the functioning of the algorithm.

We start by creating N equal arrays of frequencies starting from `w_start` and ending with `w_end` with step `w_step`. For the input signal $u(t)$, we define the amplitude `Amp` and the frequency `w0`, whereas for the noise we define the variance `var`. For every combination of the natural frequencies, after defining the state matrix, the input and output matrices of the network and its state-space representation, we compute the SNR value through nested for loops and we store it in `SNR_now`: in particular, if the value of `SNR_now` at the step i is bigger than the value of `SNR_before` at the previous step $i-1$, the code updates the value of `SNR_before` and saves in `w_SNR_max` the

frequencies at the step i. At the end of the nested for loops, the variable `w_SNR_max` returns the set of natural frequencies that maximizes the SNR. Please note that the choice of the input and output nodes is crucial and it might happen that the selected output node is not reachable by the selected input node: consequently, particular attention must be given to the choice of the topology as well as to the choice of input/output nodes.

To conclude, the code proves that, when we have a network of E-I oscillator whose interconnection pattern is based over an acyclic graph, in which the output node is actually reachable by the input node and the input signal is characterized by a single, sufficiently high frequency:

- the network is always stable when the single nodes are stable, since the set of eigenvalues of the former is exactly given by the union of all the eigenvalues of the single nodes;
- the SNR (with which we characterized the quality of the propagation of the signal along the network) is maximized through the synchronization of the nodes on the same frequency, which is exactly the frequency of the input signal.

6.5 General Interconnection Pattern and Future Research

Expanding the optimization problem to networks of E-I oscillators with more general topologies appears to be quite difficult and cumbersome to approach from an analytical and theoretical point of view: in fact, when considering, for example, directed cyclic graphs or, more generally, random graphs we must deal with topologies that may give rise to an unstable network. As we outlined in Section 6.2, the computation of the eigenvalues of the network state block matrix is hardly solvable by hand or analytically, unless we resort to simpler configurations, as we did with the DAG configuration or the series configuration, which can be obviously seen as a particular case of the former.

Along with the stability of the network, the optimization problem we defined and presented throughout the previous sections might turn out to be even more difficult to solve then the stability problem. In fact, we encountered an analogous complexity when we attempted to solve the optimization problem for the positive-feedback interconnection (which basically is the simplest case of a cyclic graph) and the final results turned out to be bounded either by the stability conditions on the parameters of the E-I oscillators and by the several simplifications that were carried out along the analysis.

However, we still had the chance to find some interesting numerical evidence. In particular, when dealing with high frequency signals, the optimal value of the SNR is numerically achieved when all E-I systems are synchronized with one another, i.e., they have the same natural frequency, or very close to be synchronized. In particular, the parameters of the single nodes must be chosen with special attention, in order to preserve the overall stability of the network: in fact, dealing with higher values of γ with respect to the corresponding gains of the oscillators may help to stabilize the network, whereas lower values of γ might hinder its overall stability, thus originating an unstable network. This constraint reflects what was outlined in the positive-feedback configuration, where the parameter γ was actually bounded from below by a quantity that was strictly correlated to the gains of the single oscillators.

In order to numerically solve the optimization problem for general topologies, we suggest using the same code that was presented and outlined in the previous section. The code can be found in Appendix E along with some comments on its functioning and some functions that automatically generate particular

types of adjacency matrices, such as strictly triangular or random ones.

To better understand the validity and implications of these numerical results, future studies are needed, which may possibly give some theoretical evidence on why the synchronization of brain areas is beneficial for information transmission. In particular, a first fundamental step may consist in providing analytic conditions for the stability of E-I systems interconnected through graphs with cycles.

Conclusion

The present thesis aimed at investigating the interplay between communication and synchronization in neural networks, as well as at giving a brief and concise insight into the wide and intricate world of neural models. Based on both analytical and numerical results, it can be concluded that the synchronization is actually a crucial factor in improving communication and information transmission through different brain regions. The results suggest that, in order to enhance the communication and thus the propagation of signals along neuronal clusters and networks, the synchronization between different brain regions is needed, if not essential.

Modelling neural regions with the simple and effective E-I model we proposed in this paper helped us examining many different topologies and interconnection patterns and for each of them we found clear evidence that brain regions synchronize with one another in order to strengthen and enhance information transmission along the neural network. Moreover, the E-I model we proposed grounds itself on the main assumption that communication throughout the network is carried out mainly by excitatory connections: we showed how the synchronization among the oscillators is preserved thanks to this particular feature, providing also an idea on why neural networks show a net preponderance of excitatory connections over inhibitory connections.

Although the analysis we carried out throughout the work was influenced by some simplifications as well as by few (strong) assumptions, the results we highlighted are remarkably interesting and they might pave the way the future studies and research on the significantly crucial role of synchronization in communication-oriented networks, such as the neural networks.

Appendix A

Graph theory

In this chapter we will introduce some basic definitions and key concepts of graph theory that will be helpful in order to analyze the more general case where different neural oscillators are interconnected in a generic network.

A.1 Undirected graph

Definition A.1.1 (Graph)**.** A graph, also called undirected graph, consists of a set $\mathcal{V}$ of nodes and of a set $\mathcal{E}$ of unordered pairs of nodes, called edges. For $u, v \in \mathcal{V}$ and $u \neq v$, the set $\{u, v\} \in \mathcal{E}$ denotes an unordered edge.

Am example of an undirected graph can be found in Figure A.1.

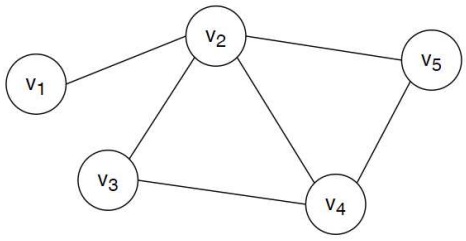


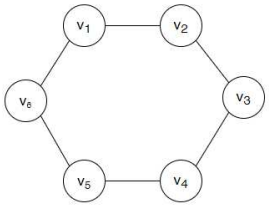
Figure A.1: Here is shown an example of an undirected graph $\Gamma = (\mathcal{V}, \mathcal{E})$ with 5 nodes. In this case $\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5\}$ and $\mathcal{E} = \{(1,2), (2,3), (2,4), (3,4), (2,5), (4,5)\}$.

Definition A.1.2 (Neighbors and degrees in graphs)**.** Two nodes u and v of a given graph are neighbors if $\{u, v\}$ is an undirected edge. Given a graph Γ, let $\mathcal{N}(v)$ denote the set of neighbors of v.

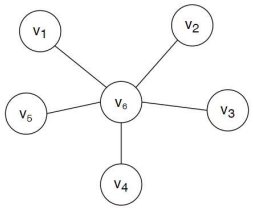
The degree of v is defined as the number of neighbors of v. A graph Γ is called regular if all the nodes have the same degree.

We can define several basic graphs of dimension n: the *path graph*, where nodes are ordered in a sequence and edges connect subsequent nodes; *cycle graph*, where all the nodes and the edges can be arranged as the vertices and edges of a regular polygon; the *complete graph* where every pair of nodes is connected by an edge.

(a) An example of a path graph.



(b) An example of a cycle graph.



(c) An example of a star graph.

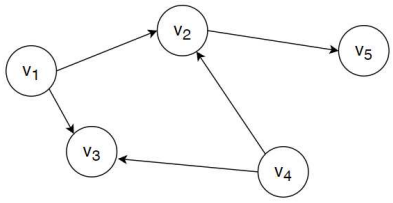
Definition A.1.3 (Path)**.** A path, or walk, in a graph Γ is an ordered sequence of nodes such that any pair of consecutive nodes in the sequence is an edge of the graph Γ.

Definition A.1.4 (Connected graph)**.** A graph Γ is said to be connected if there exists a path between any two nodes.

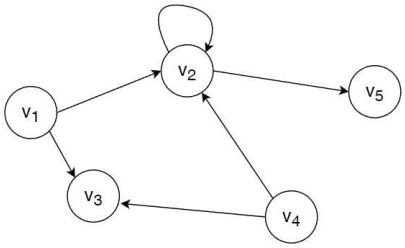
A.2 Directed graph

Definition A.2.1 (Directed graph)**.** A directed graph, also called digraph, of order n is a pair $\Gamma = (\mathcal{V}, \mathcal{E})$, $\mathcal{V}$ is a set of n elements called nodes and $\mathcal{E}$ is a set of ordered pairs of nodes called edges. For $u, v \in \mathcal{V}$, the ordered pair (u, v) denotes an edge from u to v. A digraph is undirected if $(u, v) \in \mathcal{E}$ anytime $(v, u) \in \mathcal{E}$. In a directed graph a self-loop is defined as an edge from a node to itself.

An example of a directed graph, or digraph, of order 5 is shown in Figure A.3a. In Figure A.3b a graph with a self-loop is shown.



(a) An example of a directed graph of order 5, meaning with 5 nodes.



(b) An example of a digraph with a self-loop edge on node v_2.

The definition of *path* can be extended for a directed graph: a *directed path* is an ordered sequence of nodes such that any pair of consecutive nodes in the sequence is a directed edge of the directed graph.

A directed graph Γ is called *strongly connected* if there exist a directed path from any node to any other node, *weakly connected* if the undirected version of the digraph is connected.

In a directed graph with an edge $(u, v) \in \mathcal{E}$, u can be called an *in-neighbor* of v, while we refer to v as the *out-neighbor* of u.

We can also extend the definition of *degree* for a generic node of a directed graph, introducing the in- and out-degree. In particular, the *in-degree* $d_{in}(v)$ and *out-degree* $d_{out}(v)$ of a generic node v are the number of in-neighbors and out-neighbors of v, respectively. A graph is defined as *topologically balanced* if each node has the same in-degree and out-degree. Moreover, every node with in-degree equal to 0 is called a *source*, while every node with out-degree equal to 0 is called a *sink*.

For example, in Figure A.3a, v_1 is the source of the graph, since it does not have any in-neighbor, while v_5 is the sink of the graph, since it does not have any out-neighbor.

A.3 Weighted graphs

We could also be assigning a weight to the edges of the directed graph in order to obtain a so-called weighted graph.

Definition A.3.1 (Weighted graph)**.** A weighted graph is defined as $\Gamma =$

$(\mathcal{V}, \mathcal{E}, \{w_e\}_{e\in\mathcal{E}})$, where the pair $(\mathcal{V}, \mathcal{E})$ is a directed graph and where $\{w_e\}_{e\in\mathcal{E}}$ is a set of strictly positive weights for the edges $\mathcal{E}$.

An example of a weighted graph can be found in Figure A.4.

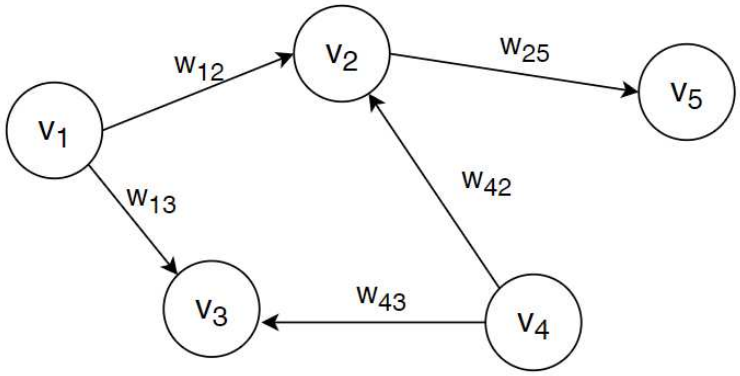


Figure A.4: An example of a weighted graph, where a weight w_{ij} is associated to each edge (i, j).

A digraph, or unweighted digraph, $\Gamma = (\mathcal{V}, \mathcal{E})$ can be perceived as a weighted graph where all the weights associated to the edges are set to 1, hence $w_e = 1$ for all $e \in \mathcal{E}$.

The concepts of connectivity and the definitions of in-neighbors and out-neighbors defined for the directed graph are still valid for a weighted directed graph. We can extend the concept of in-degree and out-degree of a generic node v by summing the weights of each in-neighbor and out-neighbor of v, respectively. Therefore, we will then obtain a *weighted in-degree* and a *weighted out-degree*. If each node as the same weighted in-degree and out-degree, the weighted graph is said to be *weight-balanced*.

A.4 Adjacency matrix

In order to represent in a more concise way the graph and its interconnections we will now introduce the so-called *adjacency matrix*.

Definition A.4.1 (Adjacency Matrix)**.** Given a weighted directed graph $\Gamma = (\mathcal{V}, \mathcal{E}, \{w_e\}_{e\in\mathcal{E}})$, the weighted adjacency matrix of Γ is the $n \times n$ non-negative matrix A where each entry (i, j) of A is equal to the weight $w_{(i,j)}$ of the edge $(i, j) \in \mathcal{E}$ and all the remaining entries are set to zero.

For an unweighted digraph $\gamma = (\mathcal{V}, \mathcal{E})$ the adjacency matrix A is defined as

$$a_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases}$$

For an undirected graph the adjacency matrix A is a binary matrix which is also symmetric. If there are no self-loops all the elements on the diagonal of A, i.e. a_{ii} for $i = 1, \ldots, n$, are set to zero.

For example, let us consider the graph presented in A.1, the adjacency matrix $A_{undirected}$ associated to the undirected graph is the following:

$$A_{undirected} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix},$$

where we can notice that the matrix is symmetric, all the values are either 1 or 0 and the values on the main diagonal are set to zero.

The adjacency matrix $A_{weighted}$ associated to the weighted graph shown in Figure A.4 is

$$A_{weighted} = \begin{bmatrix} 0 & w_{12} & w_{13} & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{25} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & w_{42} & w_{43} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Since the labels of a graph may be permuted without changing the underlying graph being represented, there are in general multiple possible adjacency matrices for a given graph.

The relationship between a graph and the eigenvalues and eigenvectors of its adjacency matrix is studied in spectral graph theory and the set of eigenvalues of the adjacency matrix is usually called *graph spectrum*.

In case of an undirected graph, the adjacency matrix is a real symmetric matrix and therefore orthogonally diagonalizable, meaning that its eigenvalues are real algebraic integers.

The *adjacency spectrum* of some basic graphs, such as a path graph or a cycle graph, can be computed in closed form for arbitrary dimension n.

Appendix B

LIF model

```
clear all;close all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% GOAL:
% Verify the relationship of the firing rate with constant
% injected current and compare the theoretic firing rate
% with the spike count rate for different values of the
% injected current
% Hypothesis:
%->single trial
%->constant input current
%->neuron = 'leaky' capacitor C_m + resistance R_m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% MODEL PARAMETERS
dt = 0.1;           %time step [ms]
t_end = 1000;       %total time of run [ms]
t_StimStart = 100;  %time to start injecting current [ms]
t_StimEnd = 800;    %time to end injecting current [ms]
V_rest = -70;       %resting membrane potential [mV]
V_th = -55;         %voltage threshold [mV]
V_reset = -75;      %reset voltage [mV]
V_spike = 20;       %spike voltage [mV]
R_m = 10;           %membrane resistance [MOhm]
tau = 10;           %membrane time constant [ms]

% VECTORS
t_vect = 0:dt:t_end;                    %time vector
```

```
V_vect = zeros(1,length(t_vect));       %voltage vector
V_plot_vect = zeros(1,length(t_vect));  %voltage plot vector

I_Stim_vect = 1.5:0.05:1.6;
%magnitudes of injected current[nA]

% Integrate LIF model equation
% tau*dV/dt = -(V - V_rest) + R_m I_e

plot_num = 0; %number of plots

for I_Stim = I_Stim_vect
    %for-loop over different I_Stim values
    i = 1; %loop index
    plot_num = plot_num + 1;
    V_vect(i) = V_rest;% set the first value of V equal to V_rest
    V_plot_vect(i) = V_vect(i);% if no spike, plot actual voltage V

    I_e_vect = zeros(1,t_StimStart/dt);
    %from 0 to t_StimStart
    I_e_vect = [I_e_vect, I_Stim*ones(1,1+((t_StimEnd-t_StimStart)/dt
       ))];
    %from t_StimStart to t_StimEnd
    I_e_vect = [I_e_vect, zeros(1,(t_end-t_StimEnd)/dt)];
    %from t_StimEnd to t_end

    num_spikes = 0;

    for t=dt:dt:t_end
        % for-loop through all values of t
        V_inf = V_rest + I_e_vect(i)*R_m;
        V_vect(i+1) = V_inf + (V_vect(i) - V_inf)*exp(-dt/tau);

        if (V_vect(i+1) > V_th)
            %cross threshold value
            V_vect(i+1) = V_reset;
            V_plot_vect(i+1) = V_spike;
            num_spikes = num_spikes + 1;
            %count the number of spikes
        else
            %did not cross threshold value
            % -> plot the actual voltage
```

```
            V_plot_vect(i+1) = V_vect(i+1);
        end
        i = i + 1;
    end

    av_rate_vect(plot_num) = 1000*num_spikes/(t_StimEnd - t_StimStart
       );
    %average firing rate [Hz]
    % note: 1000 because we treated time in [ms]

    figure(1)
    subplot(length(I_Stim_vect), 1, plot_num)
    plot(t_vect, V_plot_vect)
    subtitle(['I_e = ',num2str(I_Stim),' mA']);
    if plot_num == 1
        title("LIF model with constant input current I_e")
    end
    if (plot_num == length(I_Stim_vect))
        xlabel("time [ms]")
    end
    ylabel("V [mV]")
    ylim([-80, 25]);
    grid on

end
%saveas(gcf,'different_current.png');

% firing rate obtained from computing the reverse
% of the interspike interval

I_th = (V_th - V_rest)/R_m; %current threshold: below this value ->
    no spikes
dI = 0.001; %[nA]
maxI = 2; %[nA]
I_vect = I_th:dI:maxI;

r_isi = 1000./(tau*log((R_m*I_vect+V_rest-V_reset)./(R_m*I_vect+
    V_rest-V_th)));
% note: 1000 because we treated potential in [mV] and current in [mA]

figure(2)
plot(I_vect,r_isi,'k')
```

```
hold on
plot(I_Stim_vect,av_rate_vect, 'rx')
title('r_{isi} — r_{ave}')
xlabel('Injected current [nA]')
ylabel('Firing rate [Hz]')
legend('r_{isi}','r_{ave}',"Location","best")
hold off
%saveas(gcf,'t_isi.png');

figure(3)
plot(I_vect,r_isi,'k');
xlabel('Injected current I_e [nA]');
ylabel('firing rate [Hz]');
title('r_{isi}');
xlim([1.4,maxI]);
%saveas(gcf,'r_isi.png');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Appendix C

Firing rate estimation

[Main Program]

```
close all;clear all;clc;

%% Parameters
N = 10;          % number of trials
T = 20;          % time window rectangular kernel [ms]
sigma = 2;       % variance Gaussian kernel
alpha = 0.8;     % parameter of alpha function
rate = 10;       % Poisson rate
time = 1;        % time interval of each trial [ms]

[spikeMat, tVec] = poissonSpikeGen(rate, time, N); % tVec = vector of
    simulation times [ms]

% Time Vectors
dt = 0.01;
t1 = -100:dt:100;
t2 = 0:dt:100;
x = -length(tVec):length(tVec);
y = 0:length(tVec);

%% Kernel Function
rect = [zeros(1,length(tVec)) ones(1,T) zeros(1,length(tVec)-T)]/T; %
    Causal rectangular kernel defined on interval x
gauss = exp(-(x/(sigma)).^2/2)/(sigma*sqrt(2*pi)); % Gaussian kernel
   defined on interval x
```

```
cexp = [zeros(1,length(tVec)) alpha^2*y.*exp(-alpha*y)]; %Alpha
    function defined on interval y

%% Plot the Kernels
gauss_t = exp(-(t1/(sigma)).^2/2)/(sigma*sqrt(2*pi));
cexp_t = [zeros(1,length(t2)-1) alpha^2*t2.*exp(-alpha*t2)];

figure(1)
plot(t1,cexp_t,'k','linewidth',1.5);
xlabel('Time [ms]');
title('Alpha function')
xlim([-2,10]);
ylim([0,0.35]);
%saveas(gcf,'alphafunction.png');

figure(2)
plot(t1,gauss_t,'k','Linewidth',1.5);
xlabel('Time [ms]');
title('Gaussian kernel')
xlim([-10,10]);
ylim([0,0.25]);
%saveas(gcf,'gaussiankernel.png');


%% Firing rate estimation

z = zeros(N,length(tVec));
z2 = zeros(N,length(tVec));
z3 = zeros(N,length(tVec));

for i=1:N
    spk=spikeMat(i,:); %take the ith row of the matrix
    % convolution with the rectangular kernel
    z_tmp=conv(spk,rect,'same'); % option = 'same': takes central
       part of convolution signal
    z(i,:)=z_tmp;
    % convolution with the gaussian kernel
    z_tmp2=conv(spk,gauss,'same');
    z2(i,:)=z_tmp2;
    % convolution with the exponential kernel (alpha function)
    z_tmp3=conv(spk,cexp,'same');
    z3(i,:)=z_tmp3;
```

```
end

figure(3)
plotRaster(spikeMat, 1:length(tVec));
title(['Spike trains: N=', num2str(N)]);
xlabel('Time [ms]');
ylabel('Spike signal');
xlim([-0 500]);
%saveas(gcf, 'trials.png');

figure(4)
subplot(3,1,1);
stairs(sum(z,1)/N,'k')
title('Rectangular Window');
xlabel('Time [ms]');
ylabel('Firing Rate [Hz]');
xlim([-0 500]);

subplot(3,1,2);
plot(sum(z2,1)/N,'k')
title('Gaussian Kernel');
xlabel('Time [ms]');
ylabel('Firing Rate [Hz]');
xlim([-0 500]);

subplot(3,1,3);
plot(sum(z3,1)/N,'k')
title('Alpha Function');
xlabel('Time [ms]');
ylabel('Firing Rate [Hz]');
xlim([-0 500]);
%saveas(gcf, 'trials_kernels.png');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

[**Spike Generator**]

```
% FUNCTION: poissonSpikeGen(fr, tSim, nTrials)
% Function that simulates a Poisson process
% fr = rate of the Poisson Process
% tSim = simulation time
% nTrials = number of trials
% output -> matrix of 1/0 representing the spikes for each trial
```

```
function [spikeMat, tVec] = poissonSpikeGen(fr, tSim, nTrials)
    dt = 1/1000; % s
    nBins = floor(tSim/dt);
    spikeMat = rand(nTrials, nBins) < fr*dt;
    tVec = 0:dt:tSim-dt;
end
```

%%%

[Plot Raster]

```
% FUNCTION: plotRaster(spikeMat, tVec)
% Function to translate the matrix of 1/0 into rasters of spikes

function [] = plotRaster(spikeMat, tVec)
    hold all;
    for trialCount = 1:size(spikeMat,1)
        spikePos = tVec(spikeMat(trialCount, :));
        for spikeCount = 1:length(spikePos)
            plot([spikePos(spikeCount) spikePos(spikeCount)], ...
                [trialCount-0.4 trialCount+0.4], 'k');
    end
end
ylim([0 size(spikeMat, 1)+1]);
```

%%%

Appendix D

SNR in the FBP Configuration

```
clear all; clc;

% Input Signal
A0 = 10;    %amplitude
w0 = 100;   %frequency

% Damping factor
g = 10;

% Natural frequencies
dw = 0.1;   % step
dI = 5;     % interval width
wn1 = w0-dI:dw:w0+dI; % E-I system 1
wn2 = w0-dI:dw:w0+dI; % E-I system 2

% Noise variance
var = 1;   % variance

% System parameters
b = 3;      % input conditioning
c = 3;      % output conditioning
K1 = b*c;   % System 1 gain
K2 = b*c;   % System 2 gain
```

```
% System Matrices
A1 = [];    % System 1 Status matrix
A2 = [];    % System 2 Status matrix
B = [b; 0]; % Input matrix
C = [c 0];  % Output matrix
D = 0;      % Feedforward matrix

% SNR matrix
SNR = zeros(length(wn1),length(wn2));

for i=1:length(wn1)
    A1 = [-g -wn1(i); wn1(i) -g];
    SYS1 = ss(A1,B,C,D);

    for j=1:length(wn2)
        A2 = [-g -wn2(j); wn2(j) -g];
        SYS2 = ss(A2,B,C,D);

        SYS_fbp = feedback(SYS1,SYS2,1); % closed-looop system
        H_fbp = tf(SYS_fbp);             % closed-looop system
           trasfer function
        [mag,phase,w] = bode(H_fbp,w0);  % evaluate @w0 -> magnitude

        SNR(i,j) = 0.5*(A0*mag/var)^2;   % compute the SNR
    end
end

%% Visualization of the results
imagesc(wn1,wn2,SNR);
colorbar;
xlabel('\omega_1');
ylabel('\omega_2');
%saveas(gcf,"fbp_simulation",'pdf');
```

Appendix E

Optimal Frequencies Simulation

[Main Program]

```
clear all;clc;

% NETWORK PARAMETERS
dim = 6;    % number of nodes
g = 1;      % inhibitory parameter
b = 2;      % input conditioning
c = 2;      % output conditioning

w_start = 97;    % start freqeuncy [rad/s]
w_end = 103;  % end freqeuncy [rad/s]
w_step = 1;     % step
w_dim = (w_end-w_start)/w_step;
for i = 1:dim
    wn(i,:) = w_start:w_step:w_end;  % natural frequencies [rad/s]
end

% INPUT SIGNAL PARAMETERS
w0 = 100;   % input frequency [rad/s]
Amp = 1;    % input amplitude

% WHITE NOISE PARAMETER
var = 10;
```

```
%% ADJACENCY MATRIX

% Acyclic Graph
H_adj_acyclic = [   0 1 1 0 1 0;
                    0 0 0 1 0 0;
                    0 1 0 1 0 0;
                    0 0 0 0 0 0;
                    0 0 1 1 0 1
                    0 1 1 0 0 0];

% Cyclic Graph
H_adj_cyclic = [    0 1 1 1 1 0;
                    0 0 0 1 0 0;
                    0 1 0 1 0 0;
                    1 0 0 0 0 0;
                    0 0 1 1 0 1
                    1 1 1 0 0 0 ];

H_adj_1 = createAdjTriangSup(dim);  % triangular superior
H_adj_2 = createAdjZeroOnes(dim);   % random
H_adj_3 = createAdjSymmetric(dim);  % random symmetric

H_adj = H_adj_acyclic;
% choose the adjacency matrix to be used in the simulation

%% Input and Output nodes
in_node = 1;
out_node = 4;
in_v = zeros(1,dim);
out_v = zeros(1,dim);

for i=1:dim
    if i == in_node
        in_v(i) = 1;
    end
    if i == out_node
        out_v(i) = 1;
    end
end

% Variables for the simulation
gamma = g.*ones(dim,1);     %vector of gammas
```

```
omega = w0.*ones(dim,1);    %vector of omegas
I = eye(2*dim);
Ablk = zeros(2*dim);

SNR_before = 0;
SNR_now = 0;
w_SNR_max = 0; % frequencies -> SNR_max

% Node Matrices
A = [];         % status matrix (to compute)
B = [b; 0];     % input matrix
C = [c 0];      % output matrix
D = 0;          % feedforward matrix

%% OPTIMUM FREQUENCIES
for i1 = 1:w_dim
    omega(1) = wn(1,i1);
    for i2 = 1:w_dim
        omega(2) = wn(2,i2);
        for i3 = 1:w_dim
            omega(3) = wn(3,i3);
            for i4 = 1:w_dim
                omega(4) = wn(4,i4);
                for i5 = 1:w_dim
                    omega(5) = wn(5,i5);
                    for i6 = 1:w_dim
                        omega(6) = wn(6,i6);

                        for i = 1 : dim
                        %%%%
                        % generate the diagonal matrix with status
                            matrices
                        %%%%
                        A(:,:,i) =  [-gamma(i) -omega(i); omega(i)  -
                            gamma(i)];
                        end

                        j2 = 1;
                        for j1 = 1:dim
                            %%%%
                            % generate the block status matrix of the
                                network
```

```
                    %%%%
                    Ablk(j2:j2+1,j2:j2+1) = A(:,:,j1);
                    j2 = j2+2;
                end

                Atot = Ablk + kron(H_adj,B*C);
                % network state matrix
                Btot = kron(in_v',B);
                % network input matrix
                Ctot = kron(out_v, C);
                % network output matrix
                Dtot = D;
                % network feedforward matrix

                [n,d] = ss2tf(Atot,Btot,Ctot,Dtot);
                % ss -> tf
                sys_tf = tf(n,d);
                % transfer function
                [mag, phase]  = bode(sys_tf,w0);
                % evaluate @ w0 = input fr.

                SNR_now = 0.5*(mag*Amp/var)^2;
                % compute SNR

                if SNR_now > SNR_before % if SNR now is
                    bigger -> substitute
                   SNR_before = SNR_now;
                   w_SNR_max = omega;
                end
            end
        end
    end
end
% w_SNR_max stores the value of the natural frequencies
% that let us have the maximum value of the SNR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Function: createAdjTriangSup(dim)

```
% Function: createAdjTriangSup(dim)
% creates a strictly triangular adjacency matrix
function [H_adj] = createAdjTriangSup(dim)
    H_start = zeros(dim,dim);
        for i = 1 : dim-1
            for j = i+1 : dim
                H_start(i,j) = randi([0 1]);
            end
        end
    H_adj = H_start;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Function: createAdjSymmetric(dim)

```
% Function: createAdjSymmetric(dim)
% creates a random symmetric adjacency matrix
function [H_adj] = createAdjSymmetric(dim)
    H_start = zeros(dim,dim);
    for i = 1 : dim-1
        for j = i+1 : dim
            H_start(i,j) = randi([0 1]);
            if H_start(i,j) == 1
                H_start(j,i) = 1;
            end
        end
    end
    H_adj = H_start;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Function: createAdjZeroOnes(dim)

```
% Function: createAdjZeroOnes(dim)
% creates a random adjacency matrix
function [H_adj] = createAdjZeroOnes(dim)
    H_start = zeros(dim,dim);
    for i = 1 : dim
        for j = 1 : dim
            if i == j
                H_start(i,j) = 0;
            else
                H_start(i,j) = randi([0 1]);
```

```
        end
    end
    H_adj = H_start;
end
```

Bibliography

[1] T. Akam, D. M. Kullmann, Oscillatory multiplexing of population codes for selective communication in the mammalian brain, *Nature Reviews Neuroscience*, vol. 15, no. 2, pp. 111-122, 2014.

[2] P. Dayan, L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, MIT press, 2001,

[3] E. Düzel, W. D. Penny, N. Burgess, Brain oscillations and memory. *Current opinion in neurobiology*, vol. 20, no. 2, pp. 143-149, 2010.

[4] W. Gerstner, W. M. Kistler, R. Naud, R., L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition.* Cambridge University Press, 2014.

[5] C. Hammond, H. Bergman, H., P. Brown Pathological synchronization in Parkinson's disease: networks, models and treatments. *Trends in neurosciences*, vol. 30, no. 7, pp. 357-364, 2007.

[6] A. L. Hodgkin, A. F. Huxley. "A quantitative description of membrane current and its application to conduction and excitation in nerve." *The Journal of physiology*, vol. 117, no. 4, pp. 500-544, 1952.

[7] G. La Camera. "The mean field approach for populations of spiking neurons." In *Giugliano, M., Negrello, M., Linaro, D. (eds) Computational Modelling of the Brain. Advances in Experimental Medicine and Biology*, vol. 1359. Springer, pp. 125-157, 2022.

[8] L. Lapique, "Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization." *Journal of Physiology and Pathololgy,* 9, pp. 620-635, 1907.

[9] T. Menara, G. Baggio, D. S. Bassett, F. Pasqualetti, A framework to control functional connectivity in the human brain. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 4697-4704, 2019.

[10] A. Schnitzler, J. Gross, Normal and pathological oscillatory communication in the brain. *Nature reviews neuroscience*, vol. 6, no. 4, pp. 285-296, 2005.

[11] R. Tomar, *Review: Methods of Firing Rate Estimation*, The Czech Academy of Sciences, 2019.